

XVI. ÉVFOLYAM 8. SZÁM, 1998. AUGUSZTUS, ÁRA: 588 FT

ÚJ ALAPLAP

MAGYAR SZÁMÍTÁSTECHNIKAI FOLYÓIRAT CD-MELLÉKLETTEL

Alternativa: Miért a Linux?
Kuk@coskodó: Szomorú szenzációk
Szerszámosláda: Visual Basic 6.0
Szoftverportéka: Java, vagy nem java?
Programozástechnika: Makefile "munkakerülőknék"

2

CD

A HÓNAP TÉMÁJA:

REDUNDANCIA

Ha én egyszer kinyitom a számat...!

Akkor azt lehet, hogy meghallják. Ha viszont azt akarja, hogy híre biztosan eljusson az ország minden szegletébe (vagy akár a Föld túlsó felére is), konferenciáit, sajtótájékoztatóit, kiállításait több százezren látogassák, akkor sugározzon élő közvetítést az Interneten!

Az Elender Internet élő, helyszíni közvetítést ad képen és hangban bármiről, aminek Ön igazán hangot akar adni. Az Ön és cége kész és élő videóanyagait hozzáférhetővé teszi a hálózaton, országos és nemzetközi nyilvánosságot biztosít, webrádiót üzemeltet és home page tervező stúdiója is szavát adja, hogy az esemény akkorát szól amekkorát csak lehetséges.



elender internet

ELENDER INFORMATIKAI RT.: XIII. Bp., Váci út 37. Tel.: 465-7800 Fax: 465-7899 e-mail: info@elender.hu Web: www.elender.hu

ELENDER ÜZLETEK: 1087 Budapest, Hungária krt. 8. Tel.: 210-3044 Fax: 333-4347 • 1092 Budapest, Ferenc krt. 16. Tel./Fax: 218-2858 • Debrecen, Csapó u. 61. Tel./Fax: 52/413-795

Szeged, Madách u. 15. Tel./Fax: 62/310-269 • Nyíregyháza, Nyírfátér 5. Tel./Fax: 42/405-666 • Pécs, Klimó Gy. u. 13. Tel./Fax: 72/312-820 • Szombathely, Széll K. u. 23. Tel./Fax: 94/336-479

ORSZÁGOS VISZONTLADÓI HÁLÓZAT:

Székesfehérvár, Távirat u. 18. Tel.: 22/316-763 • 24-es körzet: Szigetmonostor, Sétáló u. 14. Tel.: 20/340-282 • Salgótarján, Kassai sor 2. Tel.: 32/422-195 • Esztergom, Mátyás K. u. 11/c. Tel.: 33/331-037 • Komárom, Tancsics M. u. 3. Tel.: 34/342-888
Eger, Céhmeztérek u. 16. Tel.: 36/436-287 • Gyöngyös, Jókai u. 38. Tel.: 37/300-799 • Nyíregyháza, Nyírfátér 5. Tel.: 42/405-666 • 45-ös körzet: Kiskőrös, Víz u. 26. Tel.: 20/383-777 • Miskolc, Szent I. u. 3. Tel.: 46/340-860 • Kazincbarcika, Radnóti tér
8. Tel.: 48/318-526 • Debrecen, Csapó u. 61. Tel.: 52/413-795 • Cegléd, Pesti út 1. Tel.: 53/311-683/2 • Berettyóújfalú, Bajcsy Zs. u. 2. Tel.: 54/401-600 • Szolnok, Sütő u. 15. Tel.: 56/427-733 • Szeged, Madách u. 15. Tel.: 62/310-269 • Hódmezővásárhely,
Hóvirág u. 2. Tel.: 62/246-810 • Szenté, Petőfi u. 11. Tel.: 63/318-755 • Mohács, Vörösmarty u. 6. Tel.: 69/304-035 • Pécs, Klimó Gy. u. 13. Tel.: 72/312-820 • Kiskunfélegyháza, Kalmár J. u. 2. Tel.: 76/463-362 • Kiskőrös, Martini u. 1. Tel.: 78/312-215
Baja, Szabadság u. 26. Tel.: 79/322-633 • Kaposvár, Arany J. u. 21. Tel.: 82/420-137 • Siófok, Fő u. 196. Tel.: 84/310-817 • Tapolca, Batsányi u. 1. Tel.: 87/412-564 • Veszprém, Botev u. 1. Tel.: 88/428-235 • Szombathely, Széll K. u. 23. Tel.: 94/336-479
Győr, Corvin u. 3. Tel.: 96/319-762

INFORMÁCIÓKÉRÉS: 07 ▲

A Mikroszámítógép Magazin és az Alaplap hagyományait folytató magyar számítástechnikai folyóirat
Megjelenik havonta, CD-melléklettel

Főszerkesztő:

Faklen Pál

Szerkesztő:

Jakab Ágnes

A szerkesztőbizottság tagjai:

Ambrózy Gábor, Aszalós László,
Feleki Zoltán, Galántai Zoltán,
Herczeg József, Horlai János,
Kis János, Kovács István,
Mózes István Miklós,
Pogány Csaba, Simay Endre István,
Szondi Egon János,
Vargha Dénes, Vékony Tamás

Szerkesztőség és kiadó:

1539 Budapest, Pf. 571
VI., Dózsa György út 84/b
Telefon: 322-4417, 322-5238
Fax: 351-8015
E-mail: alaplap@mail.datanet.hu

Weblap: <http://www.alaplap.hu>

A CD-melléklet szerkesztése:

Horváth Zénó, OpenBlue Bt
1145 Budapest XIV., Bosnyák u. 1/a
Telefon: 363-5875
E-mail: zeno@openblue.telnnet.hu

Felelős kiadó:

Faklen Pál

Terjesztés:

Megyes Zsuzsanna

Hirdetésszervezés:

Árvai Katalin, Bogácsi Mária,
Galyasi Hedvig, Pap Katalin

Külföldi hirdetések:

PubliCity

Reklám- és Médiaügynökség
1537 Budapest I., Márvány u. 17.
Telefon: 156-1182 Fax: 175-3539

A kiadó a hirdetések tartalmáért és a nyomdakészen kapott hirdetések formájáért (és helyesírásáért) nem vállal felelősséget

Példányszámadatok hitelesítése:

Magyar Terjesztésellenőrző Szövetség

MATESZ

Ez a szám

9000 példányban jelent meg

Nyomtatás:

Zalai Nyomda Rt, Zalaegerszeg
Felelős vezető:

Czirkl György vezérigazgató

Terjeszti:

A Magyar Posta Rt, a Nemzeti Hírlapkereskedelmi Rt, a Hírker Rt, a Kiadói Lapterjesztő Kft és számos számítástechnikai szaküzlet

Előfizethető a kiadónál:

Új Alaplap Kiadói Kft,
1539 Budapest, Pf. 571

Bankszámlaszám:
OTP 11706016-20788599

A lap példányonkénti ára: 588 Ft
Évi előfizetési díj: 5880 Ft

Külföldi előfizetés díja:
5880 Ft + postázási költség

HU ISSN 1217-7598

Faklen Pál összeállítása	3	A HÓNAP TÉMÁJA: REDUNDANCIA	
Galántai Zoltán	5	Sapientisat	
Simay Endre István	9	A látszólagos felesleg	
Nemetz Tibor	12	A programok „túlméretezése”	⇒ *
Szondi Egon János	14	Írott nyelvek redundanciája	
Simay Endre István	15	Biztonsági tartalék	
Simay Endre István	17	Két „redundáns” számjegy	⇒ *
Simay Endre István	18	Pont hátán pont	⇒ *
		Tömörítőforrások	⇒ *
Simay Endre István	19	SZERSZÁMOSLÁDA	
		Visual Basic 6.0	⇒ *
Galántai Zoltán	20	ALTERNATÍVA	
Adorjáni Gábor	22	Miért a Linux?	
Kádár Zsolt	31	Fel az Internetre!	
		Rövid hírek az OS/2 világából	
Simay Endre István	23	CD-PORTÉKA	
		Nagyüzemi adatbáziskezelő	⇒ *
Faklen Pál	25	PRO DOMO	
		A „PR-cikk” szindróma	
Herczeg József	27	KUK@COSKODÓ	
		Szomorú szenzációk	
Ambrózy Gábor	29	SZOFTVERPORTÉKA	
		Java vagy nem Java?	⇒ *
Bánó György	33	BÖNGÉSZDE	
		HARDVERSENY	
		PALETTA	
		FOGÓDZÓ	
Dékán István	40	A Kodak hobby-kínálata	
Mózes István Miklós	51	Betűfelület, torzítás, háttér	⇒ *
		ONLINE	
Vadász Ágnes	43	Szabadalmi „mindentudó”	
		KÖZKINC	
Kovács Attila	45	HÍRHÁLÓ	
		HÁLÓZAT	
Hargitai Zsolt	46	Bővíthető rendszer	
Csató Endre	47	A csapatjátékos	
Pál Ferenc	48	Kényelmes átjárók	
Toporczy István	49	Integráló erő a hálón	
		VISSZACSATOLÁS	
		KALEIDOSZKÓP	
Lindner László	54	Tanulságos „gépi páros”	
		PROGRAMOZÁSTECHNIKA	
Simay Endre István	56	Az Allegro bemutatkozása	
Simay Endre István	58	InterDev a Webhez	
Aszalós László	59	Makefile „munkakerülőknek”	⇒ *
Álló Géza	62	Konstansok, változók, típusok	
Szondi Egon János	66	Redundancia nélkül	
		MIKROBAZÁR	
		KÖNYVESPOLC	
Vargha Dénes	69	Felégetett hidak	
Galántai Zoltán	74	DB2 for Windows NT	
		Címlapkép a Partout Media Provider illusztrációja alapján	
Feleki Zoltán	68	Karikatúrák	
		E számunk hirdetői	

FOKUSZ

ESUITE

WORKPACT

SZERSZAM

DOSWIN

LINUX

JATEKOK

OS2

WIN95

WINNT

VENDEG

IE5SETUP

FENYJAV

VB6

LAPFORG

ARCHSRC

BMPPRESS

GNU

IMGRGB

PHSULI

SUM

ESUITE

JATEK

PCR

HTML

INSTALL

RUNTIME

Fókusz

- Lotus eSuite DevPack
- WorkPlace in Action eSuite

Szerszámoszláda

- Alkalmazások DOS és Windows 3.x alá
- Alkalmazások Linuxhoz
- Játékok Linux alá
- Alkalmazások OS/2-höz
- Alkalmazások Win95-höz
- Alkalmazások NT alá

Vendégoldal

- Internet Explorer 5.0
- A Fénytörés program teljes verziója
- Visual Basic 6 (Installálásához lásd a 19. oldalon lévő cikket.)

Lapraforgó

- A Tömörítőforrások c. cikkhez (18. oldal) tartozó anyagok
- A Pont hátán pont c. cikkhez (17. oldal) kapcsolódó program
- A GNU for DOS cikksorozathoz tartozó állomány (59. oldal)
- A Pont hátán pont c. cikkhez (17. oldal) kapcsolódó program
- Képek a Photoshop sulis cikksorozat e havi részéhez (51. oldal)
- Illusztráció a lap 9. oldalán közölt cikkhez
- Cikkek a Lotus eSuite-ról

Játékvár

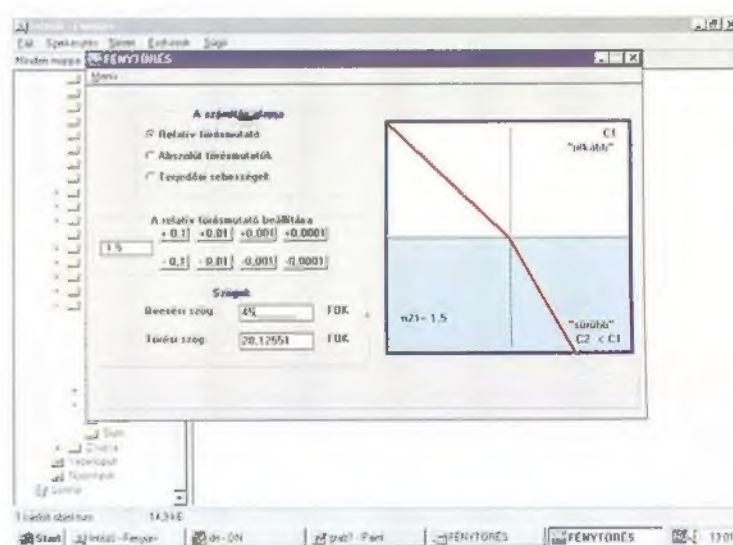
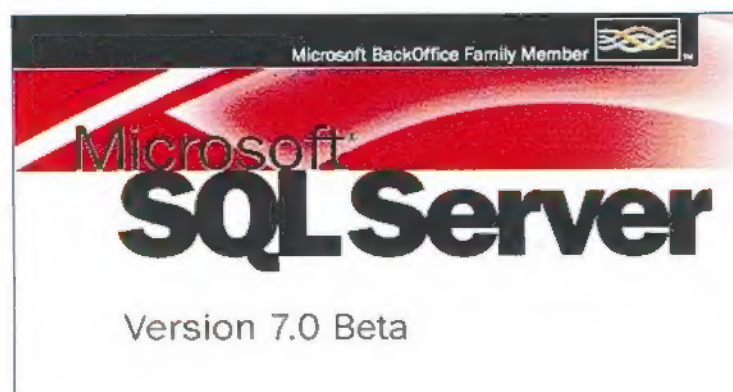
- PC Rébusz 98/8
- A CD-melléklet HTML-felülete
- Telepítőkönyvtár
- A CD-felület futtatóállományai



Az extra CD-mellékleten

Microsoft SQL Server 7.0
(Béta 3 változat)

(Használatához lásd a 23-24. oldalon lévő cikket.)



Sapientí sat

Redundancia. Milyen elvontnak és tudományosnak hangzik, pedig a legegyszerűbb emberi kommunikációnak is része. Lapozzuk csak fel O. Nagy Gábor: Magyar szólások és közmondások című könyvét a 'szó' vagy a 'beszéd' vezérszónál, és a közismert „Sok beszédnek sok az alja” közmondáson kívül még vagy tucatnyit találhatunk, amelyek az üzenet és a jel közötti ellentmondásos viszonyra céloznak, szinte kivétel nélkül elmarasztalva a felesleges ismétlést, a tartalmilag üres közlést, a szószaporítást.

Idézhetjük azonban nagy gondolkodók erre vonatkozó mondásait is, például La Rochefoucauld aforizmáját: „A nagy elmék jellemző vonása az, hogy kevés szóval sokat mondanak; a kis elmék képessége viszont az, hogy sokat beszélnek és nem mondanak semmit.” A legcsípősebb megfogalmazás azonban kétségtelenül egy nálunk kevésbé ismert (pedig Nobel-díjas) olasz költő és politikus, Giosuè Carducci (1835–1907) nevéhez fűződik: „Aki 29 szóban mondja el azt, amihez 15 is elég lenne, az egyéb aljasságokra is képes.” Azt azonban, hogy nemcsak tehetségről és morálról van szó, jól mutatja a Blaise Pascaltól eredeztetett, sokak által használt szállóige: „Meglehetősen hosszúra sikeredett ez a levelem, mert nem volt időm rövidebben írni.”

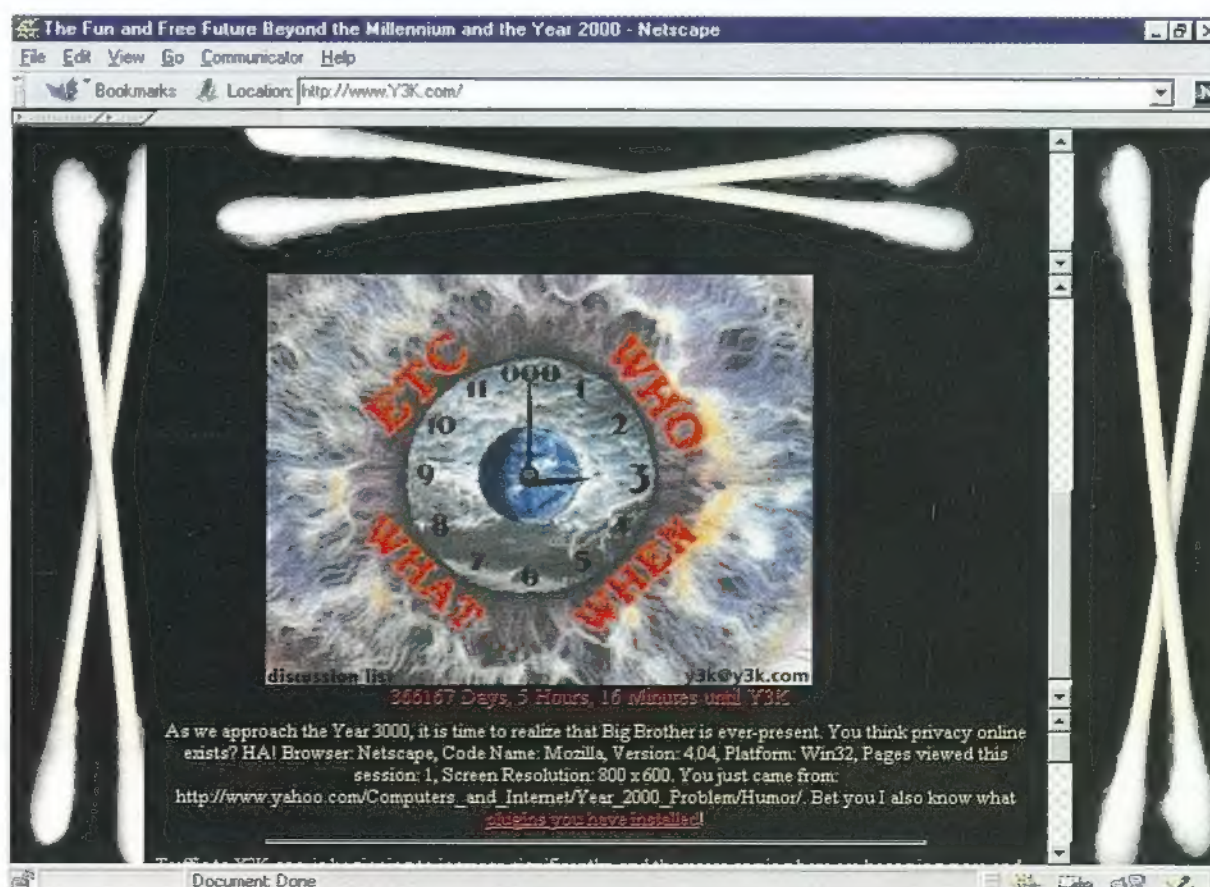
A hétköznapi kommunikációban talán éppen azért van a kritikus szellemek által feleslegesnek minősített rengeteg ismétlés, mert nem vagyunk elég tehetségesek és figyelmesek, vagy mert nincs elég időnk. A redundancia alapvető rendeltetése ugyanis, hogy biztonságossá tegye a kommunikációt, többletjeleket tartalékoljon az elkallódók helyettesítésére. És a jelek bizony úgy is elkallódnak, hogy „egyik fülönkőn be, másikon ki”. Sulykolni kell tehát, hogy valami mégis megragadjon belőle. (Lásd reklám.)


A redundancia vizsgálatának legtisztább terepe az információelmélet, amely az üzenetek átvitelének jelszükségletét vizsgálja, erősen matematikai alapokon, és bár a téma nagyon csábító, mostani összeállításunkban nem mélyedtünk el benne. (Majd máskor.)

A számítástechnikában a redundancia fogalma leginkább az adatátvitel biztonságához kapcsolódik. Jellegzetes hardveres felfogása lényegében az eszközök megkettőzését, tartalékolását jelenti. A redundancia azonban nem pótalkatrész, nem raktárkészlet és nem bankbetét. A redundancia a kommunikáció működésének folyamatosságához szükséges belső tartalék, felkészülés arra az esetre, ha a rendszerben valami „minden körütekintés és elővigyázatosság ellenére” elromlik.

Ritkábban esik szó arról, hogy a szoftverrendszerekben miért van annyi redundancia, és mi tekinthető belőlük hasznosnak és szükségesnek, illetve károsnak és feleslegesnek. Azt például valamennyien naponta tapasztalhatjuk, hogy döbbenetes arányban nő a programok, fájlok mérete, és nem akarunk beletörődni, hogy ennek szükségképpen így kell lennie. Talán egyszerűen csak arról van szó (Pascal után szabadon), hogy „Meglehetősen behemótra sikeredett ez a programom, mert nem volt időm Assembly-ben megírni”.

És lehet, hogy már soha többé nem is lesz?!





Delphi 4.0

**BÁRMILYEN ADATOT,
BÁRHOVÁ,
BÁRMIKOR!**

**Delphi 4.0 a legnagyobb
teljesítményű
RAD fejlesztőeszköz
legújabb változata**

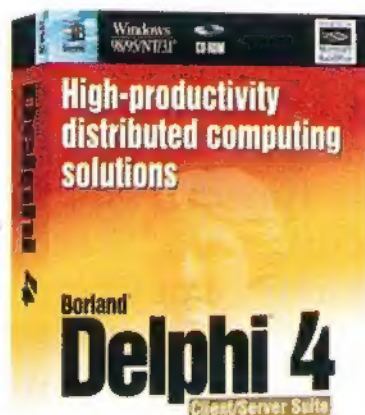
A Delphi 4.0 Client/Server számos újdonságot tartalmaz, többek között:

- Testreszabható fejlesztői környezet, dokkolható "toolbarokkal",
- "Code Explorer" egyszerűbb navigáció a forráskódban,
- "Project Group Manager" egy project több alkalmazásból is állhat,
- Microsoft Transaction Server támogatás,
- Teljes CORBA szerver és kliens támogatás,
- Visigenic Object Request Broker,
- Oracle 8 támogatás

Július 15. előtti megrendelés esetén az upgrade árból 3 % kedvezmény.
Bemutatóra jelentkezés: info@borland.hu

Borland
Magyarország
an Inprise Company

Borland Magyarország, 1143 Budapest, Hungária krt. 79-81., telefon: 252-8145
Fax: 252-8773, internet: <http://www.borland.hu>, e-mail: info@borland.hu




FAST AV MASTER '98

gyorsan megADAT ik !

Seagate AV Professional

Cheetah 10.033 rpm, 3.5" Ultra SCSI HDD
Barracuda 7.200 rpm, 3.5" Ultra SCSI HDD
Ultra2 SCSI 18Gb AV HDD-k
a világ leggyorsabb lemezei!

1012 Budapest, Várkút utca 9.
Tel/Fax: 214-8621, 214-8623 E-mail: sales@adi.hu

a választás lehetősége...

Apple brother CITIZEN Canon EPSON FUJITSU HP HEWLETT PACKARD MINOLTA NEC OKI olivetti Panasonic SAMSUNG star Tally XEROX

printer center

OTP akció

ingyenes...

Tel.: 1297-2337, 1290-646
Budapest XIII. Béke út 93.

array

Növekvő fényerő, csökkenő ár

ASK A4+	ASK A6+
750 ANSI lumen 800x600 (SVGA)	650 ANSI lumen 1024x768 (XGA)
<ul style="list-style-type: none"> • PC, Mac, audio, video bemenet • 4 x 1 Watt sztereo hangszóró • Intelligens távvezérlés • Hordtáska • Súlya csak 5 kg • 2500 órás izzó élettartam 	



Viszonteladók keresünk!
Bérlési lehetőség!

array Data Hungária Kft.
Tel.: 455-6892, 93
E-mail: array_bp@mail.elender.hu
Web site: <http://www.array.hu>

Darwin és a szoftver, avagy...

A látszólagos felesleg

Első hallásra kétségtelenül logikusan hangzik, hogy ha a más programokhoz kapcsolódva önreprodukcióra képes ártalmas kis programokat vírusoknak nevezzük, akkor az antivírus-szoftvereket az immunrendszer mintájára képzeljük el és tervezzük meg. Ez azért is vonzó ötletnek tűnhet, mert egy William Dowling nevű számítógépes szakértő még valamikor az 1990-es évek elején a matematikai logika eszközeivel bebizonyította, hogy tökéletes vírusirtó szoftver nem létezik. Tehát érdemes volna lemásolni a természet által a hasonló problémák ellen alkalmazott megoldást, az immunrendszert. Ebből elindulva már nem is olyan nehéz kiterjeszteni az önvédelemre való képesség gondolatát az élettelen rendszerekre.

Miért is ne tekintsük az egész számítógépes rendszert legalábbis nagyon hasonlónak az élő organizmushoz, kérdezik néhányan a témával foglalkozók közül. Amire persze azt lehet válaszolni, hogy egy bizonyos szinten igencsak felületes hasonlóság volna ez — például akkor, ha a szoftverrendszert védő vírusirtó működése, és a szervezetet védő immunrendszer lenne párhuzamba állítva. Az életben ugyanis a betolakodóval együtt megsemmisül a már megtámadott sejt is, márpedig jobb, ha a vírusirtó program csak legvégső esetben folyamodik ehhez a megoldáshoz. És ezzel el is jutottunk az egyik alapvető különbséghez: a szoftver rendszerint nem redundáns, az élő szervezet viszont mérhetetlenül az. A bőr vagy a szív egyetlen sejtje is tartalmazza az adott teremtmény felépítését leíró, teljes genetikai állományt — ezért lehetséges például a klónozás. Amennyiben viszont elveszítünk egy szövegfájlt, akkor (hacsak nem csináltunk róla [backup] másolatot floppyra, streamerre, vagy legújabbban az Interneten ilyen szolgáltatást kínáló valamelyik cég merevlemezére), akkor az abban található adatok végleg elvesztek.

De ugyanez, a számítástechnikára nem jellemző redundancia és rugalmasság jellemző például az emberi agyra is. „Nem szabad egy agyi régióról úgy gondolkodni — írja egy modern agykutatási könyv —, mint amelynek egyetlen specifikus, mindentől független funkciója lenne... Az agy ugyan elkülöníthető anatómiai területekből épül fel, de ezek a mezők nem egymástól független miniagyak”, és ennek

megfelelően akár az is lehetséges, hogy egy végzetesen sérült rész funkcióját a többi átvegye.

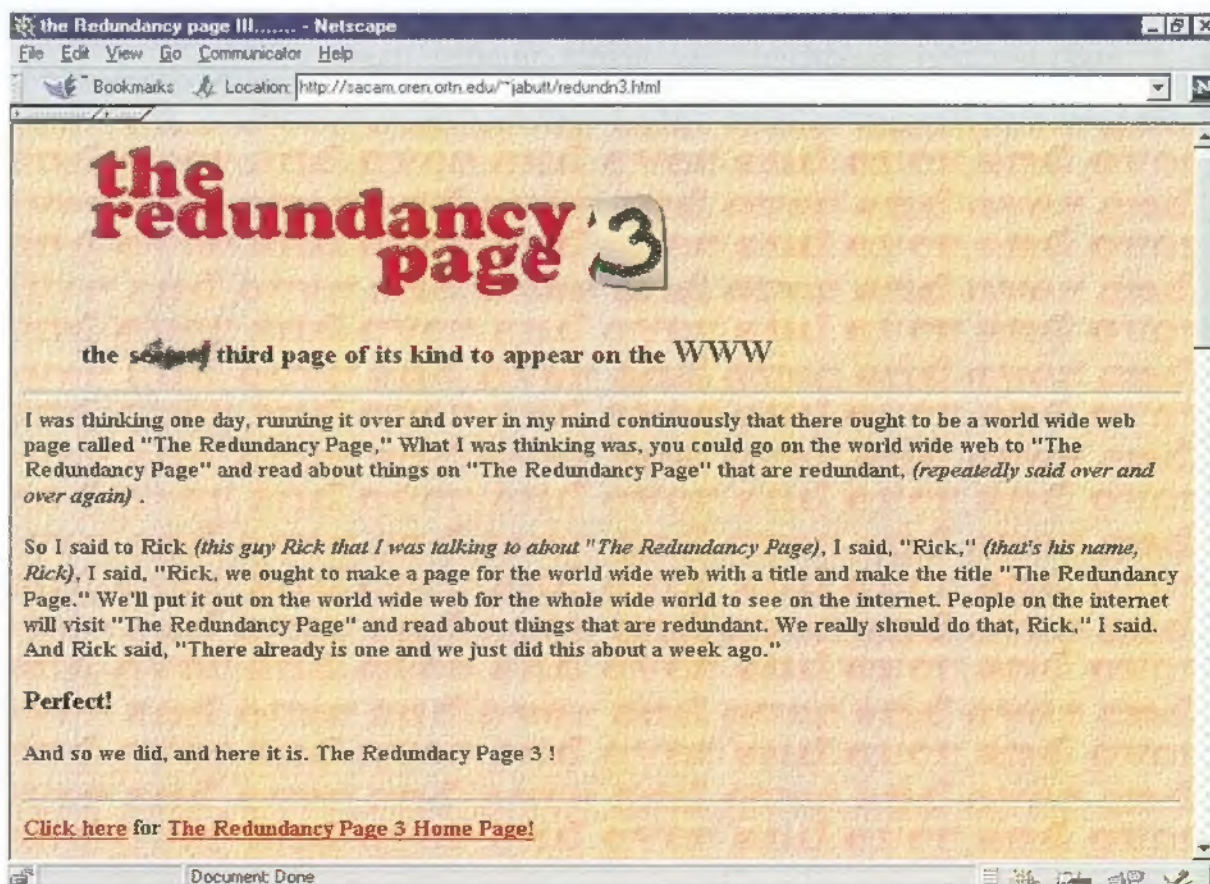
A szoftverek esetében viszont, mint tudjuk, szó sincsen semmi ilyesmiről, noha igencsak előnyös volna, ha ugyanazzal a „stabilitással” és rugalmassággal rendelkeznének, mint az agy. Azt szokás mondani, hogy amennyiben az építészek ugyanúgy dolgoznának, mint a szoftverek, akkor egy harkály is össze tudná dönteni a házakat, és ehhez azt is hozzátehetjük, hogy ha szerveztünk védelme ugyanolyan könnyen összeomlana, mint egyes széles körben elterjedt operációs rendszerek, akkor az influenzavírus már régen kipusztította volna az utolsó embert is. Ami lehet,

hogy tréfának hangzik, de azért mindenképpen érdemes eltűnődni rajta: vajon nem kellene-e mégis megpróbálnunk utánózni a természet által alkalmazott, elsőre végtelenül anyagtárolónak és redundánsnak tűnő módszereket?

A szoftver nem személygépkocsi

Sem a szoftverek, sem az élőlények — illetve általánosabban fogalmazva: bizonyos komplex rendszerek — esetében nem egyszerű a helyzet. A kanadai komputertudós, David Parnas annak idején a Reagan elnök által meghirdetett „csillagháborús” terveket tanulmányozva jutott arra a következtetésre, hogy a megfelelően összetett szoftverek (és ennek megfelelően a „Csillagháború”) programjai is igencsak sajátosan képesek viselkedni, mivel alapvetően kétféle komplex rendszer létezik.

Az egyik folyamatos (continuos), a másik szaggatott (discontinuos) módon működik. A folyamatosra az autó tesztelése a példa: ha megvizsgáljuk, hogy miként fog viselkedni egy kanyarban 60 km/h, illetve 70 km/h sebességnél, akkor ebből következtetni lehet arra is, hogy mi történne, ha 65 km/h sebességgel fordulnánk be. Nem így a másik fajta, a szaggatott rendszerek esetében,





A Hungexpo Rt. új szakkiállítása

„Nem léphetünk kétszer ugyanabba a folyóba” — mondta egykoron Herakleitosz. Ez a sokat emlegetett mondás a kiállításokra is igaz, hiszen kétszer nem lehet ugyanazon a kiállításon résztvenni. Még akkor sem, ha ugyanaz a név, hiszen a változó piaci körülmények változó tartalmat igényelnek, melyhez a kiállítás szervezőjének igazodnia kell. Akkor pedig már végképp nem, ha témájában hasonló, ám mégis új kiállításról van szó.

Márpedig a Hungexpo 1999-től új nemzetközi informatikai és kommunikáció-technikai szakkiállítást indít, INFO néven.

A változásról kérdeztük dr. Iványi Lászlót, a Hungexpo Rt. kiállítási igazgatóját.

— Az idén májusban nyolcadik alkalommal zárta kapuit a Budapesti Vásárcsopontban az IFABO kiállítás, és úgy tűnik, ezúttal utoljára... Azonban a számítástechnikai cégek nem maradnak megfelelő bemutatkozási fórum nélkül, hiszen lesz INFO '99 a Hungexpo szervezésében. Mit takar ez a változás?

— A Hungexpo Rt. 1990-ben megállapodott a Bécsi Vásárral arról, hogy a Bécsben már évek óta sikeres számítástechnikai kiállítást, az IFABO-t közös szervezésben megvalósítjuk Budapesten is. Így 1991-től a Budapesti Vásárcsopontban minden évben megrendezésre került a kiállítás, melynek közös szervezésére létrehoztuk a bécsi székhelyű Expoconcept International GmbH-t. A 90-es évek elején ez jó megoldásnak bizonyult, az idő azonban a számítástechnika terén változásokat hozott. Az informatikai vállalatok összetétele az elmúlt 10 évben teljesen átalakult, s ezt egy szakkiállításnak is követnie kell. Kiállítóink 85%-a ma már magyarországi székhelyű cég, hiszen minden nagy multinacionális cég megalakította magyar vállalatát, akár hardver, akár szoftver, akár telekommunikációs cégről van szó. Ma már minden mellett szól, hogy önálló, saját szervezésű informatikai kiállítást indítsunk 1999-től.

— Mit kínál az új kiállítás az informatikai élet szereplőinek?

— Az INFO Nemzetközi informatikai és kommunikáció-technikai szakkiállítás tematikájában helyet kap a számítástechnika, a telekommunikáció, a repro- és a másolóstechnika. A számítástechnikát többé nem az irodatechnika részeként kezeljük, hanem önálló ágakra bontjuk, az irodabútor, irodatechnika területe pedig a továbbiakban nem lesz kiemelt témacsoport. Az új kihívásoknak, világtrendeknek megfelelően a rendezvényen külön hangsúlyt kapnak a multimédia eszközei, az Internettel foglalkozók, a 2000-ik év problémájára megoldást kínáló. Mint a Hungexpo minden szakkiállításánál, itt is fokozottan törekszünk arra, hogy a kiállítás magas színvonalon, megfelelő szakmai fórumot

biztosítson a résztvevőknek. A szakma képviselőiből Szakmai Bizottságot hozunk létre, akik tanácsaikkal segítik majd a szervezést. Újdonság az is, hogy Kiállítói Tanácsot alakítunk, melynek tagjai az informatikai piac szereplői által is elismert cégek lesznek az egyes fő kiállítási témacsoportok reprezentánsaiként, akik javaslataikkal segíthetik munkánkat.

Az INFO szakmaspecifikusabbá tétele — valamint az a tény, hogy a kiállítás utolsó napja egybeesett volna május elsejével — vezetett minket arra a döntésre, hogy jövőre négynapos lesz a rendezvény, az utolsó nap péntek. A nagyközönség többnyire szombati napokon tekintette meg a kiállítást, így lehet, hogy csökkenni fog a látogatók száma, ám ezzel is a rendezvény szakmaiságát szeretnénk erősíteni. A négynapos nyitvatartás természetesen nem jelenti azt, hogy kizárnánk a szakmai érdeklődésű nagyközönséget. A belépőjegyek árai számukra is megfizethetőek maradnak, s továbbra is szükségesnek tartjuk annak a rendszernek a fenntartását, miszerint az INFO-ra az iskolások csoportosan, előzetes bejelentkezéssel ingyen ellátogathatnak.

Az új rendezvénytól részeként kiállítási újságot is kiadunk a rendezvényig több alkalommal. Bevezetjük — több Hungexpo szakkiállításához hasonlóan — a látogatók regisztrálását, mely lehetővé teszi, hogy megismerjük a kiállítás iránt érdeklődők összetételét mind szakmai ágazonként, mind a döntési folyamatban betöltött szerepük szerint. Ezzel szorosan összefügg az a kezdeményezés is, hogy egy közvéleménykutató céggel látogatói és kiállítói felmérést készítettünk, ugyanúgy, mint ahogyan ez a többi kiállításainknál már évek óta gyakorlat.

— Az INFO kiállítói számára — a szakmaiság erősítésén túl — tartogat-e még más kedvező újdonságokat is a szervező Hungexpo Rt.?

— Úgy vélem, az már önmagában a kiállítókat szolgálja, hogy az ügyintézés nem Bécsen keresztül kell rendezniük. 1999-re nem emelkednek az informatikai kiállítás árai, a Hungexpo Rt. az 1998. évi IFABO osztrák schillingben vett árait számítja át forintra. Megszűnik a katalógusbeiktatási díj is, hiszen a Hungexpo kiállításainál az első katalógusbeiktatás ingyenes.

— Hogyan reagáltak az érintettek, azaz a kiállítók az INFO kiállítás hírére?

— Június közepén tájékoztattuk a szakterület 50 legnagyobb kiállítóját, valamint a legfontosabb szakmai partnereket a Hungexpo döntéséről és a változással várható előnyökről. A cégek többsége támogatta az új, hazai szervezésű kiállítás gondolatát. A legjobb visszajelzés a cégek részéről az, hogy az 1999. április 27-30. között megrendezendő INFO '99 kiállításra előzetesen kiküldött jelentkezési lapok folyamatosan érkeznek vissza és már most több mint 5000 m² területre tartanak a kiállítók igényt.

ahol azt mondhatjuk, hogy ha a szoftver autó volna, akkor nyugodtan megtörténhet, hogy egy alacsonyabb és egy magasabb sebességnél simán veszi a kanyart, a kettő közé eső értéknél viszont kivágódik és felborul. Miként az a monitor előtt ülve nap mint nap előfordul velünk.

Ilyenkor az ember nem tehet mást, mint megvárja, amíg a számítógép ismét beindul (bootol), és közben esetleg azon tűnődik, hogy miként is lesz ez majd a jövő században. Hiszen abba sem éppen kellemes belegondolni, hogy mi történne, ha egyszer egy atomerőmű programja mutatná be a „kanyareffektust”, de az sem tudható, hogy mi lesz akkor, amikor majd a gyorsliftektől a telefonokon át a bejárat ajtóig mindent szoftver fog vezérelni, vagyis az élet minden területén ugyanezzel a problémával találjuk szembe magunkat. Mert bármennyire is igaz a „There Ain't No Such Thing as a Free Lunch” mondás (azaz „olyan, hogy ingyenskaja, egyszerűen nem létezik”, mert mindenért fizetni kell), ahogyan a számítógépes szubkultúra bölcsessége tartja, az is biztos: a mikroelektronika rohamos terjedéséért fizetett ár meglehetősen magas lehet. Kevin Kelly, a Wired magazin szerkesztője például azt kérdezi, hogy „Ha a jövő... termékei egyre inkább a ma szoftverére fognak hasonlítani, akkor vajon mi vár ránk? [Szoftverhiba miatt] működésképtelenné váló tévék? Lefagyó autók? Felrobbanó kenyérpírtók?”

Az egyik megoldás a lego

Egy meglehetősen gúnyos tipológia szerint háromféle programozó létezik: a hippi, aki átlátja az egészet, miközben sem kedve, sem türelme a részleteket kódolni; a nerd (akit talán leginkább még tökfeknek lehetne fordítani), aki kiválóan elboldogul a konkrét feladatokkal, de nincsen rálátása az egészre; és végül ott van a lumpenprogramozó is. Ő már évekkal ezelőtt rájött, hogy azzal biztosíthatja a legbiztosabban a saját állását, ha mind nagyobbá és mind áttekinthetlenebbé fejleszti a cég (rendszerint egy bank vagy egy biztosítótársaság) mainframe-en futó programját.

Egy ideig már-már úgy tűnt, hogy az első két típus kihalt, a harmadik pedig diadalmaskodik, nincsen ugyanis az a (bárha mégoly zseniális) programozó, aki képes volna többmillió soros programot áttekinteni és kézben tartani. Márpedig a modern operációs rendszerek legalább ilyen nagyok, és nem kisebb program kell ahhoz sem, hogy

a számítógép képes legyen mondjuk egy repülőgépet vezérelni. Vagyis úgy tűnt, mintha lassanként kézben tarthatatlanná válnának a dolgok.

Azután a '80-as évek végén feltűnt és rohamosan elterjedt a (valójában 1969 óta létező) OOP, vagyis az objektumorientált programozás. Ennek az a lényege, hogy nem sorról sorra előre haladva, hanem modulokból dolgozunk: olyan elemekből, amelyek önmaguk működőképesek, teszteltek, és amelyeket más egységekkel kombinálva felépíthetjük és összerakhatjuk a szoftvert, mintha csak legóval játszanánk. És ez a programozás terheinek jelentős részét leveszi a vállunkról: sok különféle célra szolgáló, előre gyártott modult (sőt, modulkönyvtárakat) vásárolhatunk a szoftverfejlesztőktől, így nem nekünk kell bíbelődnünk a részle-

tekkel. És arról még nem is beszéltünk, hogy mennyivel csökkenhet ezáltal a hibák száma.

Már egy 1982-es (az IEEE Transactions of Software Engineeringben megjelent) cikk kimutatta, hogy a túlságosan hosszú, „ömlesztett” programokban összehasonlíthatatlanul több hiba szokott előfordulni, mint azokban, amelyek kisebb részekre vannak tagolva: a tanulmány egy 10 000 soros (vagyis mai szemmel nézve nem is olyan hosszú) programot említett, amelyet három alprogramra tagolva az eredeti 317 hibát 52-re sikerült csökkenteni. Az OOP esetében ráadásul azt is megtehetjük, hogy a hibás elemet egyszerűen kidobjuk, és hibátlanul illesztünk be a helyére.

Ha lemegyünk egy bizonyos mérethatár alá, akkor komoly esélyünk van

Ha a Microsoft autókat gyártana...

Még 1997 őszén, a Comdex számítástechnikai szakkiállításon Bill Gates a számítástechnikát az autógyártással állította párhuzamba, és kijelentette: „Ha a General Motors ugyanolyan ütemben fejlesztette volna technológiáját, mint a számítógépipar, akkor most 25 dollárba kerülő autókkal járnánk, melyek egy liter benzinnel 400 km-t is megtennének.”

A General Motors úgy reagált erre a megjegyzésre, hogy „Ez igaz, de ki szeretné naponta kétszer összetörni a kocsiját?” Utána 10 pontban megfogalmazta, mi mindennel járna még, ha az autógyártás is a számítástechnikai példakép fejlődési vonalát követte volna:

1. Ahányszor az úttesteken újrafestének a sávelválasztó csíkokat, mindig új autót kellene venni.
2. Az országúton gépkocsink alkalmanként minden ok nélkül leállna, amit magától értetődőnek tekintenénk, és a motort zokszó nélkül újraindítva folytatnánk utunkat.
3. Egyes manőverektől kocsink nemcsak leállna, hanem működésképtelenné is válna, és ilyenkor a motort ismét üzembe kellene helyezni (installálni). Valami különös okból mi ezt is természetesnek tartanánk.
4. Az autóba egyszerre csak egy ember férne be, hacsak nem vennénk meg hozzá a „Car95” vagy a „CarNT” terméket. Plusz a többletüléseket is meg kellene vásárolni.
5. A Macintosh napenergiával hajtott, megbízható, sokkal könnyebben vezethető és ötször gyorsabb kocsikat gyártana, de ezek csak az utak öt százalékán közlekedhetnének.
6. A Macintosh autókhoz költséges Microsoft korszerűsítőket készítenének, hogy sokkal lassúbbak lehessenek.
7. Az olajszt, a benzintank és az akkumulátor figyelmeztető lámpája helyett egyetlen jelzőfény lenne „Általános kocsihiba” felirattal.
8. Az új ülésekbe csak egyforma fenékmérettel lehetne beülni.
9. Mielőtt a légzsákrendszer működésbe lépne, feltenné a kérdést: „Biztos benne, hogy használni akarja?”
10. Aki autóbalesetbe keveredne, fogalma sem lenne arról, hogy mi is történt.

A fentiek alapján ide kívánczik Bill Gates számára egy angol nyelvterületen egyébként is népszerű mondás: „Aki üvegházban él, ne dobálózzon kövekkel!”

hibátlan programrészleteket írni, és a dolgunkat tovább egyszerűsíti, hogy a hibák rendszerint nem szétszórva, hanem egyes modulokra koncentrálódva jelennek meg. Amikor az IBM egy széria számára ilyen egységekben írta meg a kódot, akkor 420-ból 300 teljesen hibátlanak bizonyult, és a hibák nagyobbik hányada alig 31 modulra koncentrálódott. Ez persze még mindig nem valami életbiztosítás, és bár a szakértők azt mondják, hogy megfelelő energiabefektetéssel elérhető lenne a „zero defect”, azaz tényleg tudnának hibamentes szoftvert készíteni, ez azonban nem volna olcsó mulatság. És nem is biztos, hogy erre van szükség. Egy példa talán megvilágítja, hogy miért.

A Mariner I amerikai űrszondát 1962 júliusában indították útnak, ám az egyik programban valamit elírtak (az R' helyett R szerepelt), úgyhogy emiatt a földi irányítók központ szuperszámítógépe kilövés közben arra a következtetésre jutott, hogy a rakéta kiszámíthatatlannul billeg, és parancsot adott a felrobantására. Vagyis esetleg hiába dolgozunk ki akár „error free”, vagyis tökéletesen hibátlan programokat, ha az egész ilyen szinten sérülékeny marad. Ha a programozók éppen nem néznek el valamit kódolás közben, akkor is állandóan fennáll a veszélye egy hardverhibának, amely esetleg egyetlen bitet átír 0-ról 1-esre, ám a következmények így is végzetesek lesznek.

Daniel Hillis, a híres, 65 536 processzorból álló „Connection Machine” tervezője mondja egy helyütt: semmi kétsége nincs afelől, miszerint egy programozó meg tud írni egy kétmillió soros, a pilótát helyettesítő szoftvert, az

azonban nagyon is kétséges, hogy ez a programozó azután hajlandó lenne-e felülni egy ilyen gépre. Elvégre egyetlen bitnyi hiba gyakorlatilag bármikor becsúszhat (ha nem a szoftver, akkor a hardver sérülése miatt), és a következő pillanatban már zuhanunk is a föld felé.

A természetben máshogy csinálják

Megint csak Hillis szerint „Mindössze két, nagy komplexitású szerkezetekhez vezető utat ismerünk. Az egyik a mérnöki módszerek alkalmazása, a másik az evolúció. Nos, kettejük közül az utóbbi hozza létre a sokkal összetettebb dolgokat.” Utóbbiak ráadásul sokkal stabilabbak is lesznek a mérnöki úton készítettéknél. A szoftverek esetében például azért, mert azok Neumann nyomán mindeddig azon a meggyőződésen alapultak, hogy a problémák egymástól jól elkülönülő, diszkrét logikai lépésekre oszthatóak fel, ahol az egyik rész megoldás eredménye a következő bemeneti értéke lesz, és ha ezt az eljárást (alkalmasint logikai hurkokba szervezve) elég sokszor ismétljük, akkor végül megkaphatjuk az eredményt.

Egy John Holland nevű számítógépes viszont már az 1950-es években rámutatott, hogy miközben a fentebbi „soros” megoldás, addig az evolúció „párhuzamos” megoldásokat alkalmaz. Képzeljük például egy pillanatra magunk elé a DNS-t, amely miközben meghatározza, hogy miként épüljön fel egy élőlény teste, „párhuzamos komputerként” működik. Vagyis a DNS-t felépítő molekulaszekvenciák kódolás közben nem várnak arra, hogy az egyik átadja a másiknak az eredményeket, és

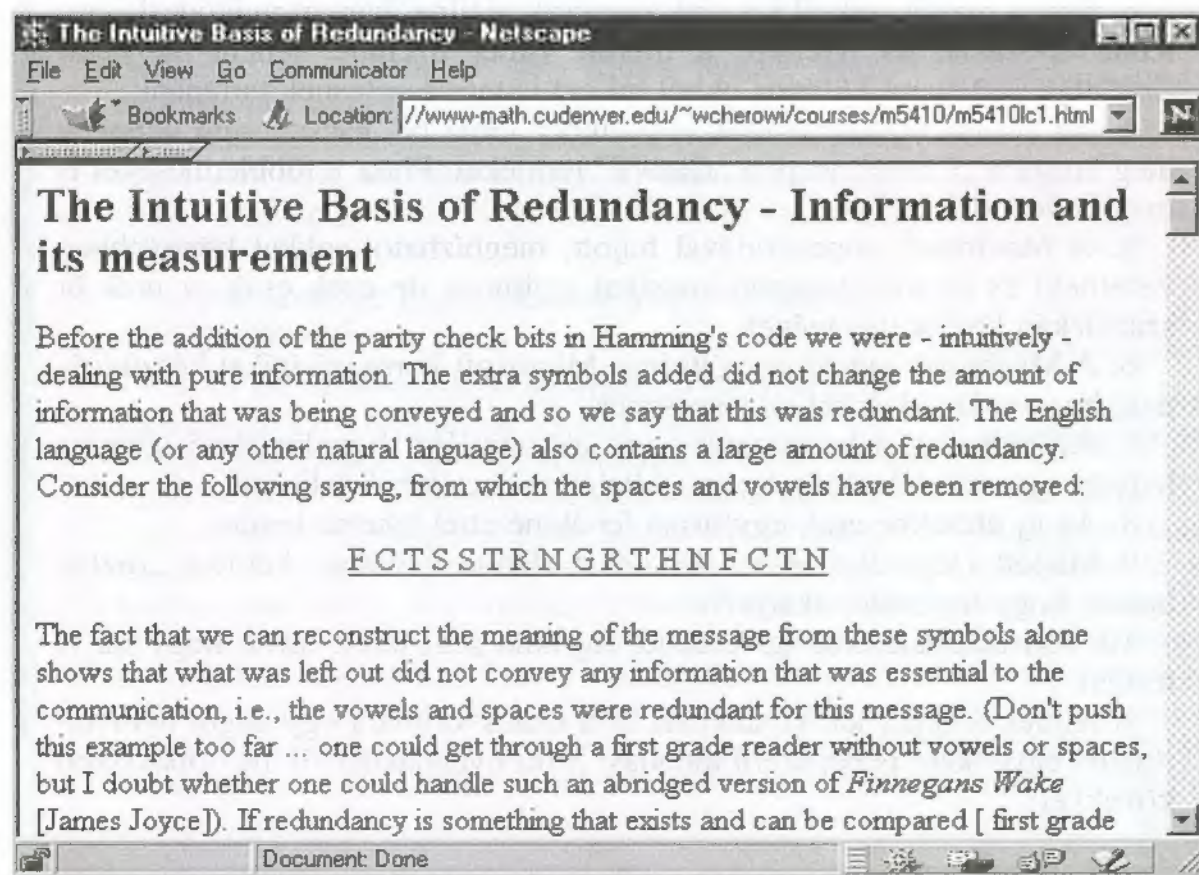
csak azután folytassák a munkát. Len Adleman 1994-ben a DNS és a párhuzamos komputer működése közötti hasonlóságot kihasználva, és bizonyos molekulaelrendeződéseket 1-esnek, másokat meg 0-nak tekintve valóban meg is építette a világ első, ún. DNS-komputerét, és ezzel meg tudta oldani többek között az ún. utazó ügynök problémáját hét elemre. A mesterséges intelligencia körében kidolgozott programok tesztelésére használt feladvány lényege, hogy adva van 7 (vagy n) város. Azt kell eldöntenünk, hogy milyen sorrendben látogassuk végig őket úgy, hogy közben a lehető legrövidebb utat tegyük meg.

Számunkra azonban a konkrét eredményeknél érdekesebb, hogy nem egy szakértő szerint a természetet utánzó, „evolúciós programozás” jelenthet megoldást problémáinkra. Legalábbis, ami a mai programok sérülékenységét illeti.

Amennyiben a jövőben nem fejleszteni, hanem „tenyésztetni” fogjuk a programokat, úgy az optimisták szerint hamarosan elérhetjük majd az élőlényekre jellemző robusztusságot is. Daniel Hillis az általa konstruált Connection Machine minden egyes processzorához egy-egy parányi programcskát rendelve hozzá, majd pedig még mesterséges parazitákat is a rendszerre eresztve azt reméli, hogy egy tulajdonképpen evolúciós folyamat végeredményeként (ahol a különböző programcskák küzdenek a „túlélésért”) minden korábbinál komplexebb — és persze stabilabb — szoftvereket lehet létrehozni. Eközben is érvényesülni fog azonban a valamit valamiért elv: az így születő megoldások nem fogják dugóhúzóba vinni a repülőgépet egyetlen bitnyi hiba hatására, mivel az élő rendszerekre jellemző komplexitással és „hibatűréssel” fognak rendelkezni. Másfelől viszont a mérnöki úton, „logikusan” megtervezett programokhoz képest szokatlanul nagyok lesznek: a 2 millió soros repülőszoftvert Hillis szerint legalább ötmillió soros, „evolúciós” programokkal lehet majd helyettesíteni.

Természetesen ez csak az első lépés. A másodikban az ezerszer nagyobb, strukturáltságukat és redundanciájukat illetően egyaránt az élő szervezetekre emlékeztető programok következnek, amelyek majd minden korábbinál komplexebb feladatokat tudnak megoldani. És eddig a pontig eljutva talán már tényleg érdemes lesz nekiállni, hogy az immunrendszer mintájára tervezzük meg az antivírus-szoftvert.

Galántai Zoltán



A szoftver vagy az ember tehet róla?

A programok „túlméretezése”

A programállományokat szemlélve valószínűleg mindenkiben felmerült már a kérdés, hogy tulajdonképpen mitől lesz egy futtatható fájl olyan nagy — vagy olyan kicsi. És egyáltalán szükségszerű-e, hogy akkora legyen? Ugyanazt a funkciót betöltő programok méretében 10-szeres vagy akár 100-szoros különbségek is előfordulnak. Ráadásul az exe-tömörítők, mint például az elterjedt PKLITE, ugyanazokat az állományokat sűrítve tárolják, és azok mégis közvetlenül futtathatók maradnak. Nincs itt valami felesleges redundancia?

A futtatható programállományok tömörítetlen méretét alapvetően befolyásolja, hogy melyik programnyelven írták azokat. Ugyancsak lényeges körülmény, hogy melyik operációs rendszerkörnyezetben való működésre készítették magát a programot. De sok függ a programfejlesztő eszköztől és még a programozó(k) személyétől is.

A programnyelvek tekintetében közismert, hogy legtömörebben a bináris kódot lehetne megírni. Erre jó példa az a bootolást kiváltó, 2 bájtnyi .com, amely az Új Alaplap 1997. áprilisi lemezmellékletén is megjelent. Ki-ki próbálkozhat, hogy más programozási módokon, Assemblytől a magasabb szintű nyelvekig mekkora minimális terjedelemben lehet ezt előállítani.

A skála másik végén legtöbbször az interpreteren futó programok állnak. Általában ezek a „legszellősebbek”, mivel a programutasításokat vagy szövegesen, vagy előfordítva (szövegesen, de tokenizált formában) tartalmazzák. Ugyanakkor a futtatható program méretébe itt célszerű beleérteni a futtató interpreter méretét is. Példák ezekre a DOS parancsértelmezőjén futó szövegfájlok, a különböző Basic-verziók vagy a script nyelvek.

E két szélső érték között találjuk meg a többi programnyelvet, amelyekből megfelelő fordítókkal közvetlenül futtatható gépi kódot készíthetünk. Ezen fordítók között kakukktojások bizonyos Basic-fordítók. Különösen a régebbi évjáratú DOS-os fordítók, amelyek ugyan készítettek egy .exe-fájlt, azonban ezek sajátos felépítésében megtalálható volt a tokenizált, interpretert igénylő kód, és az ugyanebbe a fájlba becsomagolt futtatóeszköz, egyfajta futásidejű parancsértelmezőként. Hasonló módon keletkezett .exe-fájl a korábbi

Visual Basic-verziókban is, azzal a különbséggel, hogy itt a parancsértelmezőt a programfájltól függetlenül létező VBxxx.DLL tartalmazta. (A Windows programokról később még lesz szó.)

A fordítók ténykedése

A többi programnyelv esetében adott az elvi lehetőség, hogy a fordítóprogram azonos utasítások végrehajtására azonos környezet számára hasonló méretű programfájlt készítsen. Ez azonban mégis így, mert a különböző fordítók különbözőképpen dolgozzák fel a szöveges forrásállományokat, vagy különböző mértékben képesek optimalizálni a fordítást a futtatható programfájl előállításánál. És ez meg nyilvánul a program fizikai méretében és memóriaigényében egyaránt, bár legtöbbször e két méretadat sem független egymástól.

A különböző fordítóprogramok közül a DOS-os világ legtömörebb prog-

ramállományait általában Assembly nyelvű fordítók állítják elő. Ezek a szöveges programforrásból első lépésben legtöbbször egy tárgykódú .OBJ-állományt állítanak elő, amelynek „legyártásához” különböző paraméterekben adhatjuk meg az opciókat. Esetenként az optimalizálás mértékét és a későbbi felhasználás céljának legjobban megfelelő formátumot is. Néhány assembler esetében rendelkezhetünk arról is, hogy az .OBJ-állomány tartalmazzon a későbbi hibakeresést szolgáló debug-információkat is. Ha a végleges program kialakításakor, az ún. linkeléskor a futtatható állományt készítő program nem takarítja ki ezeket az adatokat, amelyek a program futtatásához tulajdonképpen nem szükségesek, akkor a végleges állomány méretének jelentős részét tehetik ki a „csendes”, a futtatáskor szerepet nem játszó régiók.

Hasonló a helyzet a legtöbb C-fordítóval, még akkor is, ha a program fordítását ASM-állomány előzetes előállításával végezzük. A C nyelvű fejlesztéseknél azonban más szempontból is szerepe lehet az alkalmazott fejlesztőeszköznek, fordítónak. A piacon is számos rutinkönyvtár kapható, és ezeket bináris állományként, illetve a használathoz szükséges fejlécfájlként (.H) használja fel a programozó. Így, ha a programozó a könyvtár beépítésekor a



fejléc teljes híváskészletét, és ennek alapján a teljes bináris állományt befordítja a kész programba, jelentősen növekszik a kész program mérete. Olykor csak a fizikai méret nő a már említett debug infóval, legtöbbször azonban nagyobb lesz a program memóriaigénye is.

Ezen a redundancián legtöbbször csak a könyvtárhoz szükséges bináris állományt létrehozó programozó tudna segíteni, de bizony a neves fejlesztőműhelyek termékei között is gyakran találhatunk mérsékelten optimalizált könyvtárakat. Ezért is vált a fordítók fejlődésének egyik irányává, hogy a kész program előállításakor utólag is legyen lehetőség a gyomlálásra, részben megfelelő beállítással (tehát az említett infórészek kiemelésével), részben a nem hivatkozott változók, függvények hívásának és kódjának kihagyásával. Mivel ebben a vonatkozásban a fordítók sem egyformák, ez is oka lehet annak, hogy azonos és szabványos C-kódból is különböző méretű és memóriaigényű programokat tudunk előállítani.

Hasonló fejlődési irány jellemzi a Pascal-fordítókat, például a PC-s világban valószínűleg legerősebb Borland-féle Turbo Pascal fejlesztőkörnyezetek fordítóprogramjait is (TPC.EXE). Eltekintve a nyelvi elemkészletnek a többi programnyelvben is megfigyelhető bővülésétől, a fejlettebb verziók segítettek a fordítás optimalizálhatóságát is. Így például ki- és bekapcsolhatóvá vált a szövegváltozók hosszának ellenőrzése, amellyel párhuzamosan előre lehet definiálni a tároláshoz szükséges területet. Ez elsősorban a memóriagazdálkodásra volt jótékony hatással, mert például a Pascal-szabvány szerint egy szövegváltozó tárolásához a program 256 bájtnyi memóriaterületet használ. Így amennyiben például egy fájlnevet feldolgozó programmal dolgozunk, amelyben sok, de egyenként legfeljebb csak 12 karakteres szövegváltozó lehet, felesleges lefoglalni a memóriában további 243 karakternek is a helyet (a 0. bájt minden esetben foglalt a hosszúság tárolására). Célszerű ilyenkor a változókat STRING[12]-ként definiálni.

A Turbo Pascal fordítóknál a fejlesztés másik iránya a programozótól függetlenné tett feltételes kódbeépítés. Amíg például az első fordítók egy külső programrész (INC, TPU) teljes kódját valamennyi definiált változóval befordították a kész programba, ez az új verziókban jelentősen megváltozott. Így a jelenleg legutóbbinak tekinthető DOS-os Pascal, a 7-es Turbo Pascal

esetében a programozó már hiába definiál a kijelölt szegmens beteléréséig szinte akárhány változót, és hiába jelöl ki használatra (USES) szinte bármennyi TPU-t, a fordító csak a ténylegesen használt elemeket fogja beépíteni a kész programba. Ahogy a programozó arról is intézkedhet, hogy a kész programba ne kerüljön be debug adat még akkor sem, ha a felhasznált TPU-egységeket ezekkel fordították.

Ugyanaz pazarlóan

A különböző mértékben optimalizáltan működő fordítók megjelenése természetesen nem jelenti azt, hogy a programozó némi „jóindulattal” ne tudná megnyújtani programjának méretét vagy memóriaigényét. És nemcsak a már többször megemlített bogarászási segédlet beépítésével, hanem például az elnagyolt változóhasználattal és a kódolási konvenciókkal is. Ennek illusztrálására tekintsünk át egy egyszerű esetet. Írjunk egy rövid Pascal programot, amely két számot összead, és semmi mást nem tesz. A Turbo Pascal 7.01 fordítójával készítsünk ebből .EXE-fájlt:

```
program sum1;
var a,b,c:integer;
begin
  a:=10; b:=20; c:=a+b;
end.
```

Ezt követően tegyük strukturálttá programunkat, és az összeadásra írjunk saját eljárást:

```
program sum2;
var a,b,c:integer;
function sum(a,b:Integer):Integer;
begin
  sum:=a+b;
end;
begin
  a:=10; b:=20;
  c:=sum(a,b);
end.
```

Ha a két forráskódot lefordítjuk, azonos beállítások mellett a SUM1.EXE és a SUM2.EXE mérete rendre 1648, illetve 1680 bájt, ami jelzi, hogy az eljárás-híváshoz a fordító pluszkódot generált. Ebből következik, hogy ha akkor is különböző eljárások definiálását alkalmazzuk, amikor az fölösleges, az a kód indokolatlan megnyújtásával jár. Ugyanakkor a program áttekinthetőségét sem mindig javítja. Ha viszont ugyanazt az eljárást sokszor kellene beépíteni a programba, akkor természetesen indokolt lenne a strukturáltság elveit követni.

Ezek után jelöljük meg a programban történő felhasználásra a Turbo Pascal szabványos CRT egységét:

```
program sum3;
uses crt;
var a,b,c:integer;
function sum(a,b:Integer):Integer;
begin
  sum:=a+b;
end;
begin
  a:=10; b:=20;
  c:=sum(a,b);
end.
```

Ekkor a lefordított program, a SUM3.EXE már 3280 bájtnyira hízik. Ez mintha ellentmondana a korábban említett opcionális fordításnak, hiszen a program továbbra is ugyanazokkal a változókkal ugyanazt csinálja. De ez csak a látszat, mert az utóbbi esetben a fordító a CRT-unit előre definiált konstansait és értékkel rendelkező változóit (amelyekről például a helprendszerből tájékozódhatunk) szintén beleforgatja a kész programba. Így megfontolásra érdemes, hogy például egy általunk írt alegységben milyen konstansokat definiálunk, illetve az inicializáló részben milyen változóknak adunk értéket, vagy milyen eljárásokat hívunk meg.

Hossz és minőség

Az említettek nyilván a program memóriaigényére is hatással vannak. Erre azonban főleg a globálisnak definiált konstansok és változók esetében kell figyelemmel lennünk. Ha a program fizikai méretét figyelmen kívül is hagyjuk („végy nagyobb winchestert”), a memóriaigénytől akkor sem tekinthetünk el. Különösen nem a DOS real módjában futó programok esetében, amikor alig 640 Kbájtnyi helyen kell osztoznunk másokkal. (És Bill Gates egykori véleményével ellentétben 640 K mégsem elég mindenre.)

A változók definiálásakor nagyon káros az a programozói szemlélet, hogy a változókat aszerint definiálják, ahogy éppen szükség van rá. Például ha kell valahol egy új változó, akkor elgyalognak a program elejére. Ott definiálnak egyet, majd megírják az azt használó kódrészletet. Esetleg csak néhány sort. Ilyenkor az új változót és annak memóriaterületét fordításakor a program a teljes működési időre lefoglalja. Valószínűleg ez is oka lehet annak, hogy különböző programozók egyazon probléma megoldását szolgáló termékei között olykor jelentős az eltérés.

Érdemes elgondolkozni azon, hogy hátha érdemesebb a csak egyes eljárásokban használt változókat helyileg definiálni, lokális élettartammal és memóriafoglalással. Vagy némi többletidőt ráfordítva, típusonként néhány dzsóker-

ként használható változóval kiváltanánk a csak pár sorban értékkel bírót.

A programozó azonban néha rákényszerül, hogy akár mesterségesen is hozzájáruljon a program méretének és memóriaigényének növeléséhez. Az egyik ilyen ok, hogy a kész munkára igényt tartó felhasználó nem ad elég időt a forráskód és az algoritmusok optimalizálására. A másik gyakori ok, hogy a megbízó hiányos számítástechnikai ismeretei következtében úgy érzi, hogy a program hossza egyenes arányban áll a program minőségével, nemcsak a lefordított program bájtokban mért terjedelmét, hanem a gépelt programsorok számát illetően is. Ha tehát a programozót a programsorok száma alapján fizetik, akkor erős a csábítás, hogy passzív adatokkal puffasztott, esetleg több részletre szétdarabolt program készüljön.

Van, hogy a programok fejlesztésekor elindulhat egy öngerjesztő folyamat is. A kezdetben véletlenszerűen választott változónevek, a strukturált programozás jegyében megírt, esetleg következtelen névhasználati metódusok „maguk alá temethetik” a programozót. Nem emlékeztetve az alegységek pontos tartalmára, az azokban definiált változókra, a számos metódus éppen aktuális paramétereire és visszaadott értékeire, a programozó elkezd újabb változókat felvenni, újabb eljárásokat megírni. Esetleg többször is megír azonos vagy közel azonos eljárásokat, különböző nevekkkel különböző alegységekben. Ezeknek pedig a fordító bizony külön eljárás-hívásokat fog generálni, többletmemóriaterületeket lefoglalni, növelve ezzel a program belső (és ez esetben felesleges) redundanciáját. Ezen pedig csak áttekinthető tervezéssel, a már egyszer „feltalált” eljárások és programrészek nyilvántartásával, dokumentálásával — tulajdonképpen a megfelelő programozási stílus kialakításával segíthetünk. (Saját bőrön tapasztalva.)

És jött a Windows...

A programok indokolt vagy indokolatlan méretének okait kutatva még csak a DOS rendszerkörnyezetben időztünk... A windowsos programokban eleve adott a lehetőség a rugalmas memóriahasználatra és a külső objektum-, eljárás- és erőforrás-gyűjtemények (például DLL-ek) felhasználására. Ezek révén nő a programok funkcionalitása. De nő maga a Windows rendszer is. Vajon arányban van-e a megnövekedett tudás a nagyobb méretekkel? Bárki összevetheti a legújabb verziók funkcionalitását és méretét például az első

verzióéval, melyhez telepített állapotban is elég volt kevesebb mint 2 MB (Windows 1.02 = 1 876 420 bájt).

A Windows programok és különösen a 32 bites programok kevésbé érzékenyek rá, hogy mennyi a definiált változó vagy a külön könyvtárba rakott elemek száma — mindaddig, amíg a program futásának idejét ez nem befolyásolja. (Ez utóbbit is inkább a meg gondolatlan DLL-használat idézi elő.) Ugyanakkor a windowsos programoknak van egy jellemzőjük, ami jelentősen hozzájárul a fizikai méretek növekedéséhez. Ez pedig az ún. erőforrások használata.

A DOS-os programok esetében is lehetséges, hogy bináris elemként önálló állományba másoljunk össze képeket, hanganyagokat, amennyiben az egyes elemek pontos mérete ismert, és rendelkezünk a gyűjtőállományból ezeket kiolvasó, az állományokat menedzselő eljárásokkal is. Természetesen arra is lehetőség van, hogy ezeket magához az .EXE állományhoz másoljuk hozzá, akár egy copy /b utasítással. A Windows erőforrás-kezelése ezekre a megoldásokra adott rendszerszinten szabványos megoldásokat.

A Windowsban a különböző külső egységeket, amelyek itt már kész dialógusdobozok, kurzorok, szöveges listák stb. is lehetnek, meghatározott szabványos (de a 16, illetve 32 bites programokban eltérő) módon építhetjük be a program- vagy könyvtárfájlokba. Ezek az elemek mindaddig csak passzívan vannak jelen a programfájlokban, amíg futásidőben a Windows API-hívásai segítségével be nem olvassuk őket a memóriába, ahol programunk már dolgozni tud velük. Így a Windows programok esetében szinte teljesen elmosódik az összefüggés a program állományainak fizikai mérete és azok memóriaigénye között. Például nagy kép-, hang- vagy szövegállományok erőforrásként való beépítésével

jelentősen megnövelhetjük a fizikai méretet akkor is, ha ennek a memóriahasználatra alig van hatása. Ugyanakkor szabadon is hozzáférhetők olyan gyors algoritmusok, amelyekkel bináris állományokat többé-kevésbé tömöríteni tudunk, és később futásidőben kicsomagolhatunk az erőforrás-gyűjteményből.

A futásidejű kicsomagolásra erőforrásoktól függetlenül, DOS alatt is van példa. Ezek az EXE-tömörítők általában úgy működnek, hogy a tényleges programfájlokat (az .exe, illetve .com fájlakat) egy tömörítőprogrammal összenyomják, majd pedig az önkicsomagoló tömörítvényekhez hasonlóan elé tesznek egy kicsomagolót. A különbség, hogy nem a merevlemezen bontják ki a programot, hanem a memóriában, miután elindításakor az oda betöltődött. A „röptömörítő” ezután átalakul egyfajta kernelprogrammá, és kihasználva, hogy a DOS 21h megszakításának van olyan szolgáltatása, amely lehetővé teszi a futtatás kezdeményezésének átirányítását a kicsomagolt program kezdőcímére, lefuttatja az eredeti programot.

A fentiekben elmondottak talán érzékeltetik, hogy igazán mindig csak „belülről” lehet látni, hogy egy feladat megoldására készített program mérete szükségképpen akkora-e, amekkora. Az esetleges indokolatlan redundanciák kiküszöbölésének lehetőségeit vizsgálva nem szabad megfeledkezni arról sem, hogy a méretek csökkentésének túlzásba vitele esetleg a programozás határfokának, a program áttekinthetőségének, sőt futásbiztonságának rovására mehet. Valószínűleg ritkán fognak készülni minden szempontból optimális, hibátlan számítógépes alkalmazások. De ennek lehetőségét soha nem szabad teljesen elvetni, akkor sem, ha megvalósítása számos gyakorlati akadályba ütközik.

Simay Endre István



A csonka és a teljes

Írott nyelvek redundanciája

Az alábbi írás arról szeretne áttekintést adni, hogy az írott nyelvek milyen mértékben használnak a szükségesnél több jelet a beszéd megjelenítésére, mennyire terjengősek, mennyire bővelkednek fölösleges jelekben, és ezt a bővelkedést (redundanciát) hogyan lehet mérni. Nem a tartalmi bőbeszédűség vizsgálata volt a cél, hanem az egyértelmű leíráshoz technikailag nem szükséges jelek számát akarjuk mérőszámként használni.

Egy tudományos igényességű dologtól elvárható, hogy szabatos, jól definiált fogalmakat használjon. Így egyebek között meg kellene mondani, mit is értünk íráson. Néhány éve ez még egyszerű feladat lett volna, hiszen a kutatók többsége elfogadta azt a definíciót, hogy az írás a gondolatok látható formában való kifejezése. Napjainkban ez a definíció nem engedné meg azt, hogy a digitális formában írt, elektronikus adattárolón tárolt írást írásnak nevezzük. Próbálkozhatnánk az írás eredetét követő meghatározásokkal, de itt még a szilárd kiindulópont is hiányzik: „... nincs teljesen kielégítő magyarázat a legrégebbi betűírások keletkezésére nézve sem” [Varga Géza: Bronzkori magyar írásbeliség, Írástörténeti Kutató Intézet, Budapest, 1993].

A grammatikai szabályokat használó definíciók sem alkalmasak az előttünk álló feladat alapos vizsgálatára. Ezért mi az információelméletből kölcsönzünk egy, a céljainknak jobban megfelelő definíciót, bár ez az első pillantásra a valóságtól kissé elrugaszkodottnak látszik.

Tekintünk egy véges jelkészletet, amit ábécének, és aminek elemeit betűknek nevezünk. Írásnak nevezzük ezen betűk egy tetszőleges véletlen sorozatát, amely bizonyos — általunk nem teljesen ismert — sztochasztikus törvényeknek felel meg. Ez a definíció a beszéd fonetikus megjelenítését absztrahálja. Lehetővé teszi — legalábbis elméleti szinten —, hogy várható értékekről beszéljünk. Ereje abban rejlik, hogy ezek az elméleti várható értékek a gyakorlatban nagy pontossággal megbecsülhetők. E definíció mellett a „bővelkedés” mérését felcserélhetjük a tömöríthetőség mérésével. Megkérdezhetjük, hogy egy adott hosszúságú írás minimálisan hány bittel kódolható úgy, hogy a kódolt bitsorozatból az

eredeti betűsorozat egyértelműen visszaállítható legyen, miközben a kódot még a konkrét írás („individuális betűsorozat”) ismerete előtt kell megadni. Mindezt a kérdést globálisan kell felvetni, egy kódnak „átlagosan” kell jónak lennie.

Információelméleti alapokon

E kérdés már megválaszolható az információelmélet egyik legfontosabb fogalma, az entrópia segítségével. A zajtalan kódolás alaptétele éppen ennek egy számunkra hasznos tulajdonságát állapítja meg: a leggazdaságosabb egyértelműen dekódolható kódok átlagos szóhossza lényegében azonos a forrás entrópiájával. Az információelmélet megalapításánál Shannont a II. világháború alatt kriptográfiai, szövegtitkosítási vizsgálatok motiválták. [Shannon, C. E.: A Mathematical Theory of Communication, Bell System Techn. Journal, 27(1948), 379–424, 623–657.] Az egyetlen elméletileg is megfejtethetetlen titkosíráshoz ugyanakkora véletlen kulcsra van szükség, amekkora szöveget titkosítani akarunk. Ha tehát sikerül egy továbbítandó szöveget feleannyi, harmadannyi jellel átírni, akkor a szükséges véletlen jelek száma is ennek megfelelően kevesebb lesz. Természetes volt tehát a kérdés, hogy mennyire lehet tömöríteni egy szöveget.

Az, hogy jelentős mértékű tömörítés lehetséges, már régről ismert volt a kriptográfia gyakorlatából. Morse ügyvédje például a kereskedők titkos szótára címmel publikált egy típusszöveggyűjteményt („utasítom, hogy vegyen számomra a Name részvényekből x darabot”). Az ebben szereplő mondat-töredékekhez rövid, 2–4 betűs kódokat rendelt hozzá, amivel nagymértékű tömörítést, és ezáltal jelentős táviratkölt-

ség-megtakarítást ért el. Ehhez magyar példaként megemlíthető az Osztrák–Magyar Monarchiában használt három-jegyű rejtjelkulcs.

Az információelmélet arra keres választ, hogy hány bittel írható le egy szöveg tartalmi változtatások nélkül, míg a redundancia azt méri, hogy hány bitet takaríthatunk meg. Meg kell tehát mondanunk, mihez képest takaríthatjuk meg ezeket. Ezt azon bitek számával mérhetjük, ahány bittel az ábécé valamennyi betűjét le tudjuk írni. Ehhez viszont az ábécé nagyságát kell meghatároznunk. Angol ábécé esetén 2-szer 26 kis- és nagybetű van + 10 számjegy, és ehhez jönnek még a szokásos írásjelek, a pont, vessző, szóköz stb., számítsunk ezekre 14 lehetőséget. Ez 76 „betűs” ábécét jelent, ami 52 darab 6 bites és 24 darab 7 bites kóddal jeleníthető meg, és átlagosan 6,316 bitet jelent. Ehhez képest azonnali megtakarítást jelent a szövegtitkosításban kialakult évszázados gyakorlat.

A rejtjelezés során nem tesznek különbséget a kis- és nagybetűk között, elhagyják az írásjeleket, a számokat betűkkel írják ki. Tehát angol nyelv esetén marad a 26 betű és a szóköz, amihez 5 darab 4 bites és 22 darab 5 bites kódszó kell, átlagosan 4,815 bit. Ez az egyszerű átírás máris kb. 25 %-os redundanciáról tanúskodik.

Nyelvünk esetében

Természetesen merült fel az igény, hogy az entrópiavizsgálatokat a magyar nyelvre is el kell végezni. Első lépésként számítógépes feldolgozással, nyelvstatisztikai táblázatok segítségével kerestünk felső becslést az írott magyar szövegek entrópiájára. Ezeket az eredményeket Szilléry Andrásval közös dolgozat tartalmazza. [Alkalmazott Matematikai Lapok, 5(1979), 69–87.]

A vizsgálat idején elérhető számítástechnikai szintnek megfelelően a betű- és betűpár-gyakorisági táblázatok készítésével a következő entrópiabecslések adódtak:

Nulladrendű entrópia: $H(0) = \log 31 = 4,95$. Csak betűgyakoriságok alapján az elsőrendű entrópia $H(1) = 4,40$. Betűpár-gyakoriságok alapján a másodrendű entrópia $H(2) = 3,70$.

Régi tapasztalat, hogy a magánhangzók és az írásjegyek elhagyásával adódó redukált szöveg nagy biztonsággal rekonstruálható. Heurisztikusan ez azt jelenti, hogy a valódi szöveg entrópiája közel van a mássalhangzókra redukált szöveg entrópiájához. Ezért indokolt a lényegesen kisebb ábécéjű mássalhangzós szövegek statisztikai vizsgálata. Ez a redukció lehetővé tette, hogy betűhármas-statisztikát is készítsünk. Ilyen redukált szöveg entrópiájára a statisztikai becslés az alábbi eredményezte:

$$H(3) = H(\text{redukált szöveg trigrammjai}) = 2,23$$

Tekintettel arra, hogy a mássalhangzók 0,513 relatív gyakorisággal fordultak elő az eredeti szövegben, a harmadrendű entrópiára nyert becslés alapján heurisztikusan adódó entrópiabecslés:

$$H(3) = H(\text{magyar újságszövegből}) = 1,65$$

Pusztán statisztikai úton nem várható elfogadhatóan jó becslés az írott szövegek entrópiájára. Ezért Shannon már az információelmélet bölcsőjénél szubjektív becslési eljárást javasolt. [Shannon, C.: Prediction and entropy of printed English, Bell System Technical J. 30(1954), 50–64.]

Shannon és a „Szerencsekerék”

A Shannon's Guessing Game néven elterjedt módszer lényege a következő:

Egy játékvezető felolvas egy véletlenszerűen választott szövegből n számú egymás utáni betűt. A kísérleti személy (játékos) a soron következő betűre rákérdezéssel tippel, mindaddig, amíg ki nem találja ezt a betűt. Vele szemben az az elvárás, hogy legjobb tudása szerint igyekezzen a lehető legkevesebb kérdéssel célhoz érni. Ez az elvárás azt jelenti, hogy a megelőző rész ismeretében a soron következő betűre mindig annak feltételes valószínűségei szerint, azok csökkenő sorrendjében kérdezzen rá.

A játékvezető feljegyzi, hogy az egyes betűket hányadik kérdésre sikerült kitalálni. A kérdésszámok gyakorisági eloszlásának entrópiája adja a nyelv entrópiájának Shannon-féle szubjektív becslését. Shannon becslésének magyar nyelvre történő végrehajtása során a nyert becslések 1,15 és 2,8 között változtak.

A Cover–King-módszer

Shannon szubjektív entrópiabecslési módszere a soron következő betűk gyakorisági sorrendjének megtippelését várta el a kísérleti személyektől. Ezt továbbfejlesztve Cover és King 1978-

ban javasolt egy új módszert. [Cover, T.M., King, R.C.: A Convergent Gambling Estimate of the Entropy of English, IEEE Trans. on Info. Theory, 1978.]

Gambling-technikának nevezett módszerük Shannonéhoz hasonlóan a nyelvet jól ismerő kísérleti személyek (fogadók) nyelvismeretét használja ki. Induláskor a fogadó egy adott tőkével rendelkezik. Mint a Shannon-játékban, itt is a szöveg valamely részének ismeretében kell tippelnie a következő betűre. Tippelése most azt jelenti, hogy valamilyen megosztásban a teljes tőkéjét fel kell tennie az ábécé betűire. Megadja, hogy aktuális tőkéjének hányadrészét teszi fel az egyes betűkre. Nyereménye a realizált betűre tett tétjének és az ábécé méretének a szorzata.

A fogadót arra kéri: úgy játsszon, hogy szétosztási arányai egyezzenek meg a feltételes valószínűségekkal. Ekkor tőkéjének „átlagos” növekedése az entrópia konzisztens becslését adja. Ha a szétosztási arányok eltérnek a feltételes valószínűségektől, akkor az adódó becslés felső korlátot határoz meg.

A módszer írott magyar nyelvű szövegekre történő alkalmazását Szászné Simon Judittal közösen végeztük. [Magyar Nyelv, LXXXV(1989) No.4, 427–438.] Az entrópiabecslések 1,13 és 1,49 között váltakoztak.

Entrópiabecslés csonkított szövegekkel

A Shannon-féle „Guessing game” és a Cover–King-féle „Gambling” technika alkalmazása rendkívül időigényes, ezért csak viszonylag rövid szövegrészeket lehet megvizsgálni, ami nem eredményez kellően meggyőző következtetéseket. További hátránya, hogy kisszámú kísérleti személyre korlátozó-

dik. Ezért szükségesnek láttam egy olyan módszer kidolgozását, amely akár tömegmérésekben is elvégezhető, és egy megbízható, elfogadhatóan közeli felső korlát megállapítását teszi lehetővé. [Entropy estimation via reconstruction of mutilated texts, Publications de Colloque International du CNRS}, Cachan, (France) 1977, 389–397.]

Módszerem lényege a következő:

Tekintsünk egy írott szöveget és egy csonkítási szabályt. A szabálynak megfelelően betűket hagyunk ki a szövegből. Például kihagyhatunk minden második betűt, vagy minden betűről 1/2 valószínűséggel kisorsoljuk, hogy kihagyjuk-e. Ez utóbbi esetben kihagyhatjuk a kisorsolt betűket nyomtalanul, de meg is jelölhetjük, hogy honnan hagytuk ki őket. Ha a megtartott betűkből álló sorozat (megcsonkított sorozat) alapján sikerül rekonstruálni az eredeti szövegeket, akkor az eredeti szöveg entrópiája megegyezik a megcsonkított szöveg entrópiájával, és a csonkított szöveg entrópiáját statisztikai úton felülről becsülhetjük.

A magyar nyelvre vonatkozó konkrét kísérleteket nyolcféle csonkítási szabály alapján végeztük el. A szövegek rekonstruálására 15 gimnáziumi tanulót és 15 felnőttet kértünk fel. A teljesítményekről részletesen a Magyar Nyelv 1989. 4. számában számoltunk be, ahol a felhasznált hiányos szövegeket is megadtuk.

A „legjobb” csonkításnál a szövegeket alternálva 4 és 5 betűs csoportokba osztottuk, és minden csoportból sorolással elhagytunk két betűt. Ez a szabály a magyar nyelv entrópiájára 1,61 felső becslést eredményezett.

Nemetz Tibor
nemetz@math/inst.hu



Amikor a több kevesebb, és fordítva...

Biztonsági tartalék

A cím megfogalmazásából önmagában még nem derül ki, hogy minek a biztonságáról is van szó. Ha egy atomerőmű műszerezéséről szóló szakkönyvben olvassuk ezt a fejezetcímet, magától értetődő, hogy többletműszerek beépítésével lehet fokozni az üzemeltetés biztonságát. Persze a több műszer egyben több hibaforrást is jelent: a műszer hibája az egész rendszer biztonságát csökkenti. Ezért azután 3 független mérőláncot építenek be, és megelégszenek 2 azonos értelmű jel beérkezésével („2 a 3-ból elv”). Ha folyamatszabályzó számítógépről beszélünk, mindent megduplázunk (CPU, memória stb.), remélve, hogy a gép hibája csak a felét érinti, az üzemben maradó másik fele pedig fenn tudja tartani a rendszert. De mi van akkor, ha csak a „számító” számítógépre vonatkozik a cím?

Ha a számítógépet mint gépet tekintjük, az üzemeltetés biztonsága érdekében szükség van redundanciára. Ennek eszköze például a paritásbit vagy a Hamming-kód (lásd: Új Alaplap lemez-melléklete, 1996/4). Más a helyzet viszont akkor, ha az információfeldolgozás biztonságát elemezzük.

Vegyük először szemügyre a feldolgozást, pontosabban annak programját. Ha a program jól van megtervezve, megírva, akkor minden végrehajtandó lépés csak egyszer szerepel benne. Ennek a követelménynek egyszerű a magyarázata: csak így lehet a programot biztonságosan karbantartani. (Murphy elvei szerint az azonos feladatot ellátó két programrész közül módosítás esetén az egyik biztosan elsikkad.) Ez azt jelenti, hogy a program redundanciája csökkenti a biztonságot.

Éppen ellenkező a helyzet magával az információval, vagyis az adatokkal, pontosabban szólva a programnak átadandó adatokkal. A kezelési (például billentyűzési) hibákat kizárni nem lehet, ezért az adatbevitel megtervezésénél gondoskodni kell arról, hogy olyan részadatot is be kelljen vinni, ami a többiből kiszámolható. A legegyszerűbb megoldás a beadandó összes adat összegének mint redundáns adatnak a beadása. Ezt a program is kiszámolja, majd a kettőt összeveti: szimpla adatbeadási hiba esetén az összeg nem jön ki. Többszörös hiba esetén mégis előfordulhat, hogy „jó” lesz az ellenőrző összeg, tehát ez a módszer inkább csak kevés adat esetén megfelelő. Ugyanakkor fennáll az a veszély, hogy az adatbevitelt végző személy a program hi-

bajlezésére válaszként fejben számolva „korrigálja” a hibásan kitöltött adatlapot, és ezzel megöli a teljes feldolgozást.

Ha sok vagy rendkívül fontos az adat, akkor bonyolultabb védelmi eljárást kell alkalmazni. Ilyen lehet az ellenőrző számjegy képzése. A részletek mellőzésével hasonlítsuk össze a 9-es és 11-es osztási maradék ellenőrző jegyként való bevitelét. A 9-es mellett szól, hogy fejben kiszámolható. Nem véd viszont a tipikus gépelési hiba, a számjegycsere ellen. Ezzel szemben a 11-esnél ilyen gond nincs, de hiányzik a „10-es számjegy”, amely létező maradékérték. Az ISBN esetében ilyenkor látható egy „X” az utolsó pozíción — sajnos ez a karakter nincs rajta a numerikus billentyűzeteken. (Gondoljunk egy bankkártyaolvasó terminálra!) A megoldás a „foghíjas” számsor lehet, ami persze a feldolgozás más fázisaiban gondot okozhat.

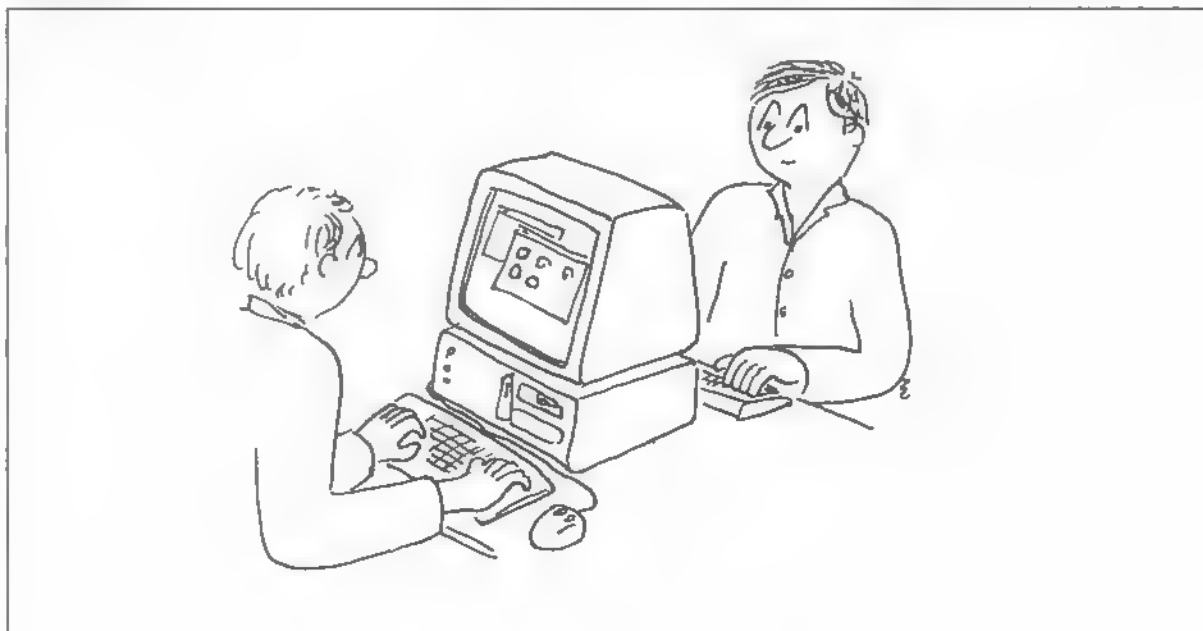
A tudományos számításokat végző, igen nagy programok inputja terjedelmes és bonyolult lehet. Ilyen esetben megéri külön programot írni, amelynek a feladata csupán a teljes adatsor beolvasása és elemzése (hihetőségvizsgálat). Megéri, hiszen éppen a főprogram bonyolultsága miatt a felhasználó nem tudja átlátni a feldolgozás teljes folyamatát, azaz nem tudja a végeredmény alapján eldönteni, hogy helyes volt-e az input. A bevezetésben már céloztunk az atomerőművek biztonságára. A biztonságos adatmegadás érdekében az egyik ilyen adatelemző program fel is rajzolja azt a geometriát, amelyet a felhasználó adatai alapján a (Monte-Carlo) főprogram felépít magának. (Az adatok a határfelületek egyenletei.) Nehéz beskatulyázni ezt a megoldást: mi redundáns benne? Talán az, hogy a grafikus output az egyenletekhez képest nem ad új információt. Ez viszont a hibakeresést hihetetlen mértékben megkönnyíti.

Végül még egy kis grafika. E-mail üzeneteinkben igen tömören tudathatjuk az olvasóval érzéseinket:

:-) Jó kedvem van
:-(Rossz kedvem van
:-() Csodálkozom

Persze itt a redundancia hiánya visszaüthet, mert ha ezekből egyetlen jel is elkallódik, rögtön megváltozhat az üzenet tartalma, vagy a jelcsoport értelmetlenné válhat.

Szondi Egon János
szondi@reak.bme.hu



Nagy ár egy kis spórolásért

Két „redundáns” számjegy

A „datumkatasztrófának” becézett 2000. évi „átállítási roham” eredete nem éppen hízog a számítástechnikusokra nézve. Az valahogy még érthető, hogy amikor az első operációs rendszerek és alkalmazások megszülettek, nagyon kellett takarékoskodni a memóriakapacitással, és kézenfekvőnek tűnt, hogy az évszám első két számjegyét meg lehet spórolni. De az már elég furcsa, hogy utána az egész szakma évtizedeken keresztül vagy nem vette komolyan, hogy a 2000. évet ezek a rendszerek nem képesek kezelni, vagy kényelmességéből egyszerűen besöpörte a szőnyeg alá a problémát, és nem volt hajlandó menet közben, folyamatosan négyjegyűvé áthangolni a rendszereket szinte a felhasználók tudtán kívül. Most viszont azok, akiket a változtatás közvetlenül érint, egyúttal szerény becslések szerint is legalább 600 milliárd dollárt fognak leszurkolni, hogy „beléphessenek” a 2000. évbe.

Az ezredforduló közeledtével egyre több cég kezdett átfogó programba, hogy szoftvertermékeit felülvizsgálja, és szükség szerint kijavítsa. Leginkább a nagy rendszerek gazdáinak kellett idejében lépniük, hogy „ne kessék le a vonatot”. A feladat mibenlétével, a megoldás módozataival és a Digital Equipment stratégiájával éppen 1 évvel ezelőtti számunkban foglalkoztunk. (Molnár Máté: A datumkatasztrófa elhárítása, Új Alaplap, 1997. augusztus.)

Ilyen átfogó koncepció a hálózati világban a Novell cég Project 2000 programja is. Ez 1996-ban indult, elsősorban az akkor forgalmazott NetWare operációs rendszernek és segédprogramjainak átvizsgálására, tesztelésére és minősítésére. A tesztelések során a kezdő dátumot 1980. január 1-jében határozták meg, mivel a legtöbb IBM, illetve IBM-kompatibilis PC esetében ez a bázisidőpont. A tesztelés során azonban nemcsak a jelen helyzetet mérték fel, hanem a módosítás és a továbbfejlesztés irányát is megadták. A probléma ugyanis nem 2000-be átlépve jelentkezik először, hanem már most is egyre gyakrabban kell a jövő évezred dátumaival számolni. Az átjárónak ráadásul kétirányúnak kell lennie, mert a mostani adatok visszamenőleges elérésére 2000 után is sokáig szükség lesz.

A Project 2000 tíz alapkritériumot fektetett le, amelyek alapján a vizsgált szoftverek minősíthetők. Így a „2000-biztos” Novell programoknak jelenleg is meg kell felelniük a következő szempontoknak:

1. Hibátlanul kell kezelniük a dátumokat egészen 2035-ig, és nemcsak az adott programon belül, hanem bármely más csatlakozó program számára korrekt módon jelölve azokat.

2. Magát a 2000. évet hibátlanul kell kezelni, amely azért is különleges, mert szökőév. (Ha egy évszám 4-gyel osztható, akkor szökőév. Ha 4-gyel és 100-zal is osztható, akkor nem szökőév, de ha 4-gyel, 100-zal és 400-zal egyaránt osztható, akkor megint csak szökőév! És ezt a programoknak kezelniük kell.)

3. Mivel az ezredfordulót követően is szükség lesz a korábbi időpontok használatára, a programoknak 1980 és

2035 között értelmezniük kell minden dátumot, a hét napjainak jelölésével együtt.

4. Szükség van a naptári időtartamok kezelésére, azaz a programnak bármely két időpont közötti időtartamot pontosan vissza kell adnia 1980. január 1. és 2034. december 31. között. Ennek ellenőrzésére felhasználhatóak többek között az alábbi kritikus napok:

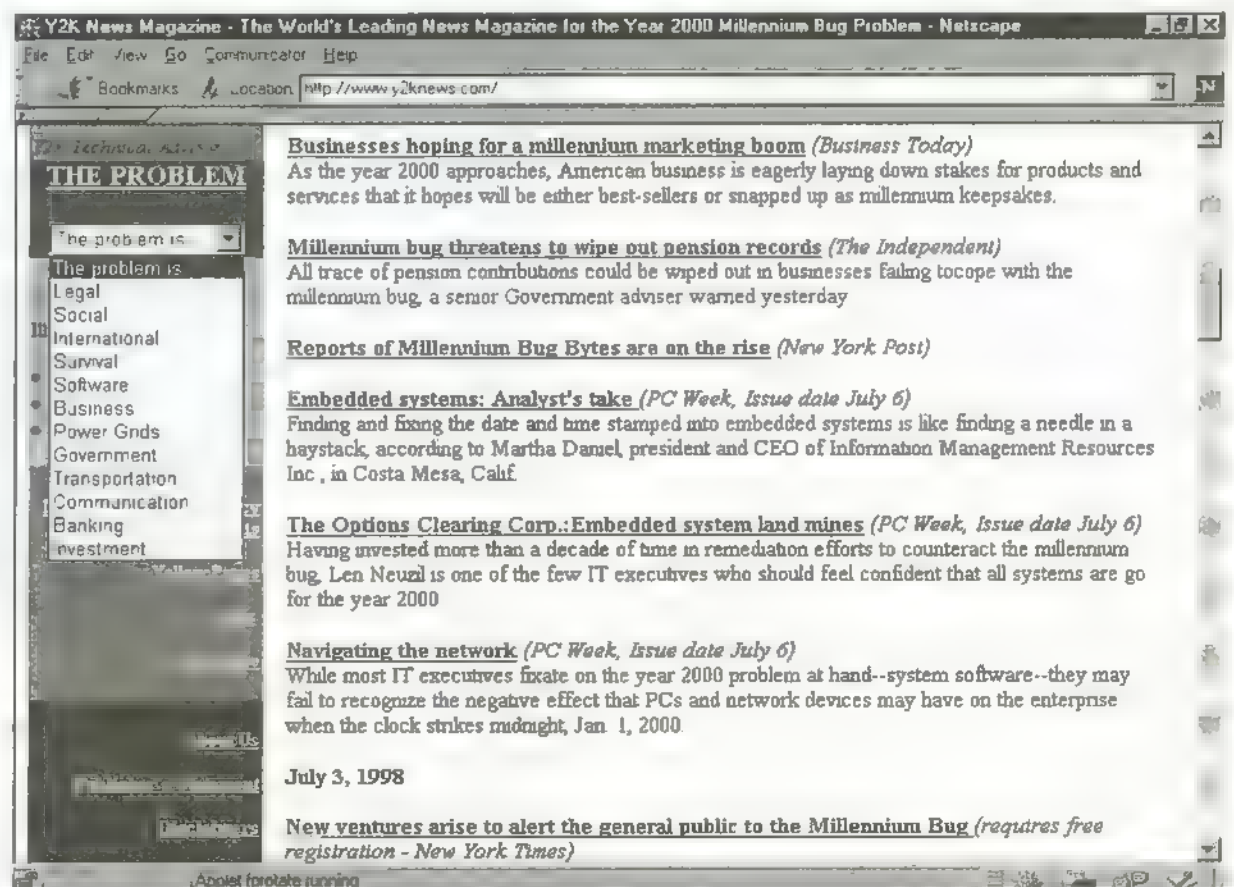
1999. december 31. péntek
2000. január 1. szombat
2000. február 28. hétfő
2000. február 29. kedd
2000. március 1. szerda
2034. december 31. vasárnap
2035. január 1. hétfő

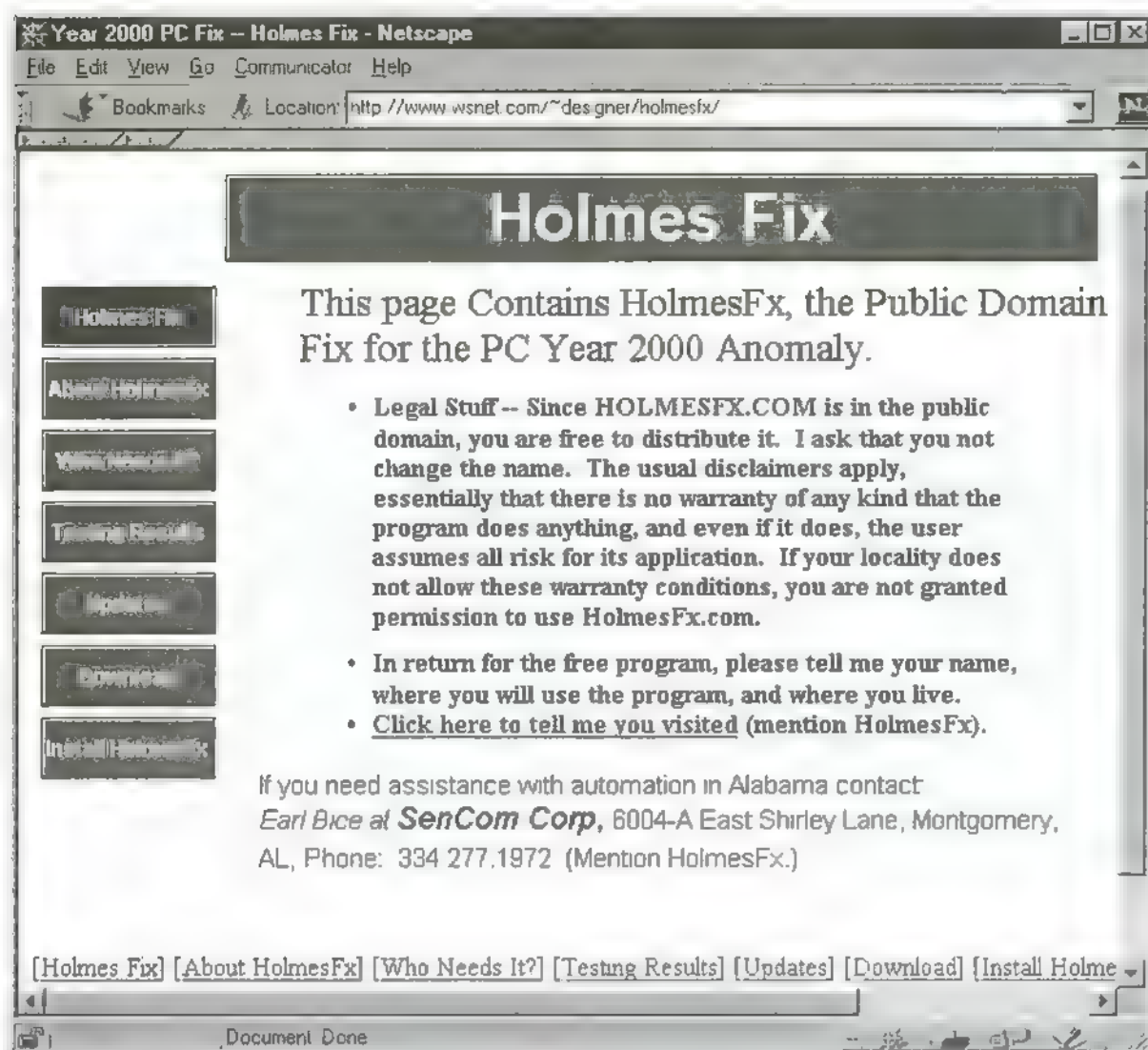
A két szélső időpont, 1980. január 1. és 2035. január 1. között az időtartam pedig 20 089 nap.

5. A programoknak, alkalmazásoknak a fenti időintervallumba tartozó dátumokkal nemcsak pontosan kell számolniuk, hanem a sorrendbe rendezést is jól kell kezelniük.

6. A kivételes eseteket és hibás időpontokat is jól kell értelmezni. Például a „no date” (nincs dátum) vagy „never” (soha) típusú adatokat is.

7. Ha a dátum és az időpont binárisan tárolódik, fokozott tesztelést igényel az, hogy a dátum ne vehessen fel negatív vagy nulla értéket. Ehhez a bármely





időpont és időtartam tárolására alkalmas változó méretet kell lehetővé tenni.

8. Bármely bemeneti fájlformátum esetén biztosítani kell a megfelelő dátumkezelést.

9. Amennyiben a bemeneti fájl konvertálásra kerül, az új formátumot előállító programnak kezelnie kell a régebbi dátummegadási konvenciókat is. A dokumentációban a fogadott és a kiadott formátumokat egyaránt le kell írni.

10. Az áttérés érdekében a fentiek alapján megváltoztatandó összes hálózati elrendezést és protokollt részletesen dokumentálni kell. Ugyanakkor az új rendszernek kliens- és szerveroldalon egyaránt kompatibilisnek kell lennie a korábbi rendszerrel, mindkét irányban kezelve a régi és új dátum- és időpont-jelzéseket.

A Novell a hálózatos operációs rendszerek és segédprogramok széles választékát forgalmazza, így a Project 2000 elindítása is jelzi, hogy a 2000. évre való átálláshoz lényeges kérdésnek tekintik a sokszor egymás mellett, több verzióban használt szoftverek megfelelőségének tesztelését. Az új verziókat és programokat természetesen már eredetileg is 2000-re felkészítve fejlesztették.

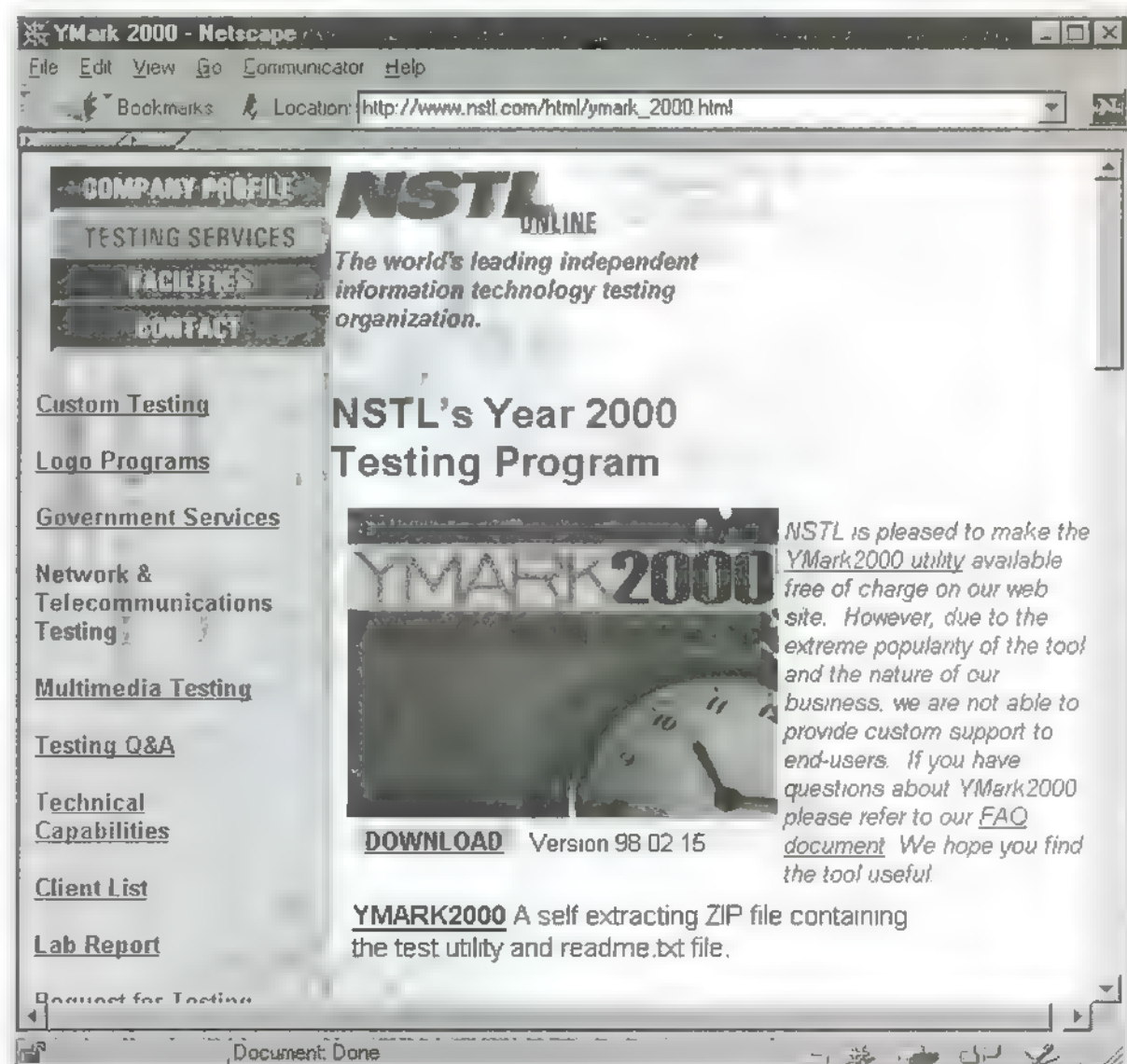
A Novell operációs rendszerei közül a nemrég béta-tesztelésre bejelentett NetWare 5.0 mellett a korábbi verziók közül a NetWare 3.2 is alkalmassá vált

az átlépésre a 2000. évbe. Ugyanakkor több más programnál adódott néhány probléma, melyek javítókészletekkel megoldhatók voltak. Ezek között egyaránt megtalálhatók egyes operációs rendszerek korábbi verziói, illetve más

hálózatos programok. Ehhez az Internetről update-készletek tölthetők le (<http://www.novell.com/p2000/patches.html>) a NetWare 3.12-es operációs rendszerhez éppúgy, mint a kliens vagy kommunikációs és biztonsági programokhoz. (Novell Clients, Novell Administrator for Windows NT v2.0c, NetWare ConnectView v2.0, BorderManager v2.1, ManageWise v2.5, Lan WorkGroup v5.0, Lan WorkPlace v5.0, Hostprint v1.11.)

Szintén hozzáférhető az update-készlet az intraNetware-hez (NetWare 4.11 Support Pack 5), amellyel a többi program irányába biztosítható a „2000-biztos” kijárat a helyi hálózatokon. Nemcsak a már említett 5.0-s NetWare alkalmas az évezredforduló korrekt kezelésére, hanem a napjainkban kapható újabb programok is. Ha pedig valaki bizonytalan, hogy az általa használt program milyen helyzetben van, a Novell honlapján rendszeresen frissített listát találhat az egyes termékekről (<http://www.novell.com/p2000/product.html>), illetve a Novelltól letölthető egy sorozatszám-azonosító segédlet, a SNIP (Serial Number Identification Program), amellyel pontosan azonosítani tudja a gépen futó NetWare-t. A sorozatszám birtokában a fejlesztőtől tudunk adatokat szerezni programunk időállóságáról.

Simay Endre István



Pixelgrafikus képformátumok

Pont hátán pont

Ugyanazt a képet más formátumúakra konvertálva azt tapasztalhatjuk, hogy a forrásfájl és a célfájl méretében nemcsak árnyalatnyi, hanem nagyságrendnyi eltérések is előfordulnak. Vajon szükségszerű ez a redundancia? És egyáltalán arról van-e szó, vagy valami másról? A vektorgrafikus és a pixelgrafikus képtárolás „filozófiájával” és annak következményeivel már többször foglalkoztunk, érdemes azonban megismerni a pixeles képekben meglévő sokféleség hátterével is.

Az információfeldolgozásnak szerves része a grafikus információk befogadása és feldolgozása. Minél részletgazdagabb maga a forrás, annál több információ észleléséről, átviteléről és megőrzéséről kell gondoskodni. Az élőlények látása ezt a funkciót igen differenciáltan és jól oldja meg.

Az élő természet látási mechanizmusából véve a mintát, a mesterséges információs rendszerek, különösen pedig a számítógépes grafika fejlődésének egyik iránya is a minél több képpontból (információegységből) felépített, a természetes képi világot színben, formában egyre jobban megközelítő képek előállítása, tárolása volt. S éppen ez jelenti az egyik szűk keresztmetszetet.

A pixelalapú képtárolás során a fájlban minden egyes képponthoz hozzárendelünk egy tárolási egységet. Általában tehát a fájl mérete a tárolt képpontok számától, a kép méretétől és felbontásától függ. Vagy mégsem?

Hardverfügő képtárolás

A legegyszerűbb megoldás a monitoron közvetlenül megjelenő kép tárolása. A PC-s világban a képmegjelenítést végző videokártya képtároló mechanizmusába megfelelő programokkal beavatkozhatunk, a videokártya memóriájából a képinformációk közvetlenül kiolvashatók és elmenthetők. Ezen alapul például a DOS-os Borland Turbo Pascal GetImage() eljárása, amely a kép dimenzióit és a képpontokhoz tartozó palettaindexeket tartalmazza. Bár az említett eljárást eredetileg a memóriába való képelmentésre találták ki, a kimentett memóriatartalom fájlba is írható.

Nagyobb probléma, hogy ez a fajta képtárolás hardverfügő, mert csak a palettaindexeket tárolja, magát a palet-

tát nem. A Turbo Pascal és a Turbo C esetében, amíg át nem kódoljuk azokat, maga a fejlesztőprogram standard palettaszíneket definiál. Jobb lesz a képtárolás, ha függetlenné tesszük az aktuális hardvertől és a fejlesztőeszköztől, mégpedig a palettának a képinformációk melletti tárolásával.

Erre többféle megoldás is született a különböző programok fejlesztésekor. Az egyik a színadatokat a képpontok adataiban való tárolása (ilyenek például a .PIX fájlok). Ezek képpontonként 3 bájtnyi adatban közvetlenül a vörös, a zöld és a kék színösszetevők (RGB) értékét tartalmazzák. Mivel pedig ezt a formátumot általában kötött méretű, 512x512 vagy 1024x1024 pixeles képeknél alkalmazták, az aktuális képméretet a fájl méretéből meghatározható.

Kötött fájl szerkezet

A másik megoldás az, hogy a képfájlban kötött formátumú rekordot tárolunk, melyben a képfelbontás, a színhasználat, a képméret egyaránt megtalálható, a képmegjelenítéshez szükséges palettával egyetemben. Ennek egyik megvalósítása a Windows-zal elterjedt, bár eredetileg nem ahhoz kitárlt DIB (Device Independent Bitmap) formátum, amely egy fejkordban tárolja a bitmap azonosítóját, a kép adatait, és ezt a 16 és a 256 színű képek esetében követik a paletta RGB-értékei. (A Windows-ban használt .BMP-nél színenként 4 bájtnyi területen.)

A képpontokat az említett színfelbontásoknál palettaindexek írják le. Ha viszont több színt használunk (például 16 milliót), akkor a fájl már nem tartalmaz palettát, viszont képpontonként igényli a 4 bájtnyi tárolási kapacitást, 1-1 bájton tárolva az RGB-értékeket, 4. bájton pedig a helykitöltő 0 értéket.

Mivel ez nem valami helytakarékos képtárolási megoldás, lényeges, hogy mindig csak annyi szín használatával mentsük el képeinket, amennyi még feltétlenül szükséges. (A BMP-fájlok méretének mesterséges hízalásáról az Új Alaplap 1997. decemberi számában már olvashattak.)

Kötött fájl szerkezetű palettatárolás valósul meg számos más képformátum esetében is: ilyen a Deluxe Paint festőprogramban bevezetett .LBM formátum és a szintén elterjedt ZSoft Paintbrush .PCX formátuma. Az utóbbi elterjedését nagyban elősegítette, hogy a megjelenésekor használatos eszközökön a 4-256 színű képeket tömörített formában tudta tárolni.

Szintén a .PCX fájlok jellegzetessége, hogy a képek RGB adatait nem külön képpontokként, hanem színsíkként tárolja. Ezzel a két megoldással, a színsíkkénti külön tárolással és a futásidejű RLE (Run Length Encoded) tömörítéssel gazdaságosan tudja tárolni a képeket. 256-nál több színt használva ez a formátum sem tárol palettát. A paletta tárolása azonban kevesebb színt használva is rendhagyó, mert a 16 színű képek palettáját a fájl elején található 128 bájt hosszú fejkord tartalmazza (a képre és a fájlra vonatkozó egyéb adatokkal együtt), a 256 színű palettát azonban a fájl végén találjuk, az utolsó 256x3 bájton (RGB).

Belső tömörítés

Időközben több más képformátumot is kidolgoztak a képpontok adatainak tömörített tárolására. Ezek egyike a .PCX-hez hasonlóan veszteségmentes tömörítést alkalmazó .GIF formátum. A GIF-fájlokban a képpontok adatait a tömörítőprogramok algoritmusaihoz hasonló eljárással sűrítve tárolják. A képfájl mérete ilyenkor nem annyira a fizikai képmérettől függ, hanem inkább a kép szerkezetétől: minél több a redundancia (azaz homogén, ismétlődő) szakasz a kép bittérképében, minél több és nagyobb azonos színű folt van a képen, annál nagyobb arányban tömöríthető a kép. Mivel a .GIF formátumnak bővítésével megvalósítható volt egyfajta képanimáció is, használatuk okkal vált általánossá a rövid letöltési időt, azaz tömör képeket igénylő internetes weblapokon. Hátránya viszont (a copyright-problémán túl) az is, hogy jelenlegi formájában maximálisan 256 színt tartalmazó képek tárolására használható.

Szintén tömörítve tárolja a képeket a JPEG-formátum (.JPG), de veszteséges tömörítési algoritmussal dolgozik.

Tömörítéskor a bittérképet egységes adatfolyamként kezeli, ezért felhasználhatósága gyakorlatilag független az alkalmazott színek számától. Nagy méretű és sok színt tartalmazó képek tárolására is jól használható. Hátránya az említett veszteséges tömörítés (ami a képi információk egy részének elvesztését jelentheti) és az, hogy a formátumleírás nem teszi lehetővé a GIF-hez hasonló animált képek készítését.

Ez utóbbi, valamint az igazán hatékony konvertálóprogramok hiánya okozza, hogy az eredetileg a GIF leváltására szánt PNG (Portable Network Graphics) nem igazán tudott elterjedni.

A felhasználói programokban emellett számos különböző, akár egyedi formátumokat is használhatunk. Annál is inkább, mert a tömörítő algoritmusok egy részének forráskódja is publikus. Ezekkel különböző tömörítési eljárásokat magunk is kipróbálhatunk.

Képeink belső redundanciájáról tapasztalati úton olyan egyszerű módszerrel szerezhetünk információkat, hogy képeinket különböző tömörítőprogramokkal tömörítjük vagy eleve belső tömörítést alkalmazó formátumokba konvertáljuk. Ilyenkor általában ugyanaz tapasztalható, amit a GIF-fel kapcsolatban már említettünk: amelyikben sok egyszínű felületből álló rész van, az jobban sűrítendő. Jelentős helymegtakarítást érhetünk el például, ha BMP formátumú képeinket PCX, GIF vagy JPG formátumban mentjük el. Az pedig, hogy azonos formátum (BMP) esetében a színekkel való takarékoság mennyire eredményes, kipróbálható akár a Windows Paintbrush programjával is.

Simay Endre István

BmpPress képtömörítő

A CD-mellékleten lévő BMPPR16 és BMPPR32 program 16, illetve 32 bites Windows-környezetre készült, forráskódja az ugyancsak közreadott UniBmp.Pas. A program által ismert képformátumok: .bmp; .dib; .gif; .ico; .wmf; .blz; .pcx; .clp; .lbm, továbbá 32 bites környezetben a .jpg is.

A képeket megkereshetjük a „Keresés” gombra kattintva, vagy rádobhatjuk a fájlt a program ablakára. Ezt követően elmenthető a tömörített kép, alapbeállításban .BLZ kiterjesztéssel, gyorsmentéssel és az eredeti névvel. A tömörítés kiterjedhet a teljes fájlra, vagy lehet bitmap-szintű, mely utóbbi esetben a fájl belső szerkezete a BMP-fájlokéhoz hasonló (ez az alapbeállítás). A beállításokat a jobb egérgombbal előhívható lebegőmenüből változtathatjuk meg.

Tömörítőforrások

A programozási feladatok egyik nagy kihívása, hogy a program által használt bináris erőforrásokat, például képeket, minél kisebb terjedelemben tudjuk mellékelni a programhoz. Kétségtelen, hogy számos tömörítőprogramot beszerezhetünk, azonban ahhoz, hogy a tömörítvényt ne csak külső program segítségével tudjuk létrehozni és a program futása során olvasni, célszerű a tömörítőegységet beépíteni a programba. Ehhez pedig az szükséges, hogy rendelkezünk tömörítő algoritmussal vagy annak a programba beépíthető és onnan futásidőben meghívható moduljával (például egy DLL-lel).

Jóval kisebb a választék, ha a szabadon elérhető programozási segédleteket tekintjük. Ezek egyike az Info-ZIP csoport produktuma. Maga az Info-ZIP csoport olyan C-programozókat egyesít, akik célul tűzték ki, hogy szabadon használható csomagoló és kicsomagoló eszközöket készítsenek minél több platformra. Az általuk összeállított programok forráskódban is hozzáférhetőek, miként a CD-mellékletre tett ZIP23A.ZIP állományban a legújabb verzió próbaváltozata is. Azt pedig, hogy az Info-ZIP forrásai milyen jól használhatók a gyakorlatban, bizonyítja a Windows Commander (C. Ghisler & Co.) beépített ZIP-kezelése.

De nemcsak ZIP-kompatibilis tömörítéshez férhetünk hozzá szabadon, forráskódban. Haruhiko Okomura és Haruyasu Yoshizaki jóvoltából dolgozhatunk olyan algoritmussal is, amelyet a LHArc 2.x tömörítő használ (LH5.PAS önálló unit, Pascal forráskódban). Itt az algoritmus alapját a Lempel-Ziv-alapú tömörítés képezi, amelyet számos más tömörítőprogram alkalmaz. (Az LZSSUNIT.PAS fájl például illeszthető a forráskódú LZ77 algoritmushoz Borland Turbo Pascal 7 unitként.)

Ugyanez az alapja a különböző LZH (Lempel-Ziv-Huffman) megoldásoknak is. Eerre komplett DOS-os példát találunk a TPLZH030.ZIP fájlban. Sajnos csak részben forráskódként, de legalább lefordított és használható Borland (Turbo) Pascal 7-es unitként. Windows-programozók szintén dolgozhatnak ezzel a tömörítési megoldással. Ehhez ízeletként szolgált az Új Alaplap 1998. májusi CD-mellékletén közreadott LZHCOMP.ZIP, amelyben a Delphi 1.0 és 2.0 verziókhoz készült tömörítő komponensből találhattunk lefordított unitfájlokat (.DCU), de forráskódként is elérhetők mind a komponens, mind a tömörítőegységet tekintve (Danny Heijl — TLZRW1.ZIP). Ezzel a Delphi-Formra helyezett komponens eljárásait meghíva, bármely fájl típus tömörítését elvégezhetjük — kétféle algoritmus alapján. Az egyik a már említett LZH, a másik a Kurt Haenen által Pascalba foglalt LZRW1/KH, mely a forráskódkészletben eredeti unitforrásként megtalálható, és más programokban is felhasználható. Így képek tömörítésére is, ahogy Herbert J. Beemster Delphi 1.0-ra készült komponense is példázza (UBITMAP.ZIP), mely tömörített fájlként képeket képes menteni és beolvasni.

Ebből a komponensből annak egy módosított formáját magam készítettem el, amely 16 és 32 bites Delphiben, illetve C++ Builderben egyaránt felhasználható (SUNIBMP.ZIP — UNIBMP.PAS). Ez nemcsak fájlszinten tömörített képeket képes kezelni, hanem a 32 bites Delphiben meglévő olyan tulajdonság is implementálva van benne a Borland által fejlesztett forráskód alapján (CLASSES2.PAS), hogy erőforrásból képes közvetlenül beolvasni képeket. Ennek beépítése a Delphi 1.0-ban végzett programozásnál feltételes fordítás segítségével történik, de a fájlszintű tömörítésen túl Windows-képek (BMP fájlok) tömörítésére is alkalmas, úgy, hogy megtartja az eredeti fejlécet, és abban jelöli a tömörítést — ezzel a hagyományos BMP-nek megfelelő fájlszerkezet megmarad.

Ahhoz, hogy ki-ki maga is használhassa a tömörített képeket, elkészült egy képtömörítő program is, amely különböző formátumokat képes fogadni, és akár bitmapszintű, akár fájlszintű tömörítéssel elmenti azokat. Alapértelmezésként .BLZ kiterjesztéssel (BmpPress — BMPPR16.EXE és BMPPR32.EXE — Windows 3.x és Windows 95, illetve Windows NT platformra).

Simay Endre István

Visual Basic 6.0

Tesztelésre — fejlesztőknek

A véletlen összjátéka, hogy miközben Könyvespolc rovatunkban, a 69–73. oldalakon részletesen foglalkozunk a Visual Basic 5-ös verziójával, a Visual Basic 6.0 éppen elkészült béta-változatát a Microsoft közreműködése révén sikerült rögtön ráaknii mostani CD-mellékletünkre. A rendelkezésre álló hely korlátai miatt azonban a telepítőkészlet nem tartalmazza a teljes Visual Studio 6.0 fejlesztőrendszert, abból csak a Visual Basic 6.0 próbaverzióját emeltük ki. (Így a CD-n lévő szoftver nem is tekinthető a Microsoft hivatalos konfigurációjának, a béta-állapotú termék kipróbálásából származó közvetlen vagy közvetett, valós vagy vélt kárért nem is vállal felelősséget.)

A Visual Studio 6.0 anyagából a Visual Basic 6.0 komponens kiemelése bemutatása azért is indokolt, mert a magyar piacon valószínűleg ez számíthat a legnagyobb érdeklődésre. A CD-mellékleten lévő telepítőkészlet csupán a VB 6 telepítését teszi lehetővé, mivel azonban a teljes „stúdió” képességeinek felvillantása is célunk volt, amennyire lehetett meghagytuk a Visual Studio 6.0 telepítőkörnyezetét, annak ellenére, hogy ezáltal a megszokott telepítési folyamatnál kissé körülményesebben juthatunk el a Visual Basic 6.0 kipróbálásához.

A Visual Basic telepítése előtt érdemes figyelmesen elolvasni az alábbi útmutatót, mert a Visual Basic csak az itt közölt lépéseket követve telepíthető a CD-mellékletéről.

A telepítés menete

A CD-ről a VENDEG/VB6 könyvtárból indítsa el a SETUP.EXE-t. Túl nagy regisztrációs adatbázis esetén esetleg figyelmeztetést kapunk annak korlátozott voltáról. Ilyenkor célszerű kilépni a telepítőprogramból, majd miután biztonsági másolatot készítettünk a SYSTEM.DAT és a USER.DAT állományokról, újraindítani a telepítést, és ekkor már a figyelmeztetés ellenére tovább folytatni azt. Ha később mégis probléma adódna, akkor az elmentett állományokból helyreállíthatjuk rendszerünknek a telepítés előtti állapotát.

Ezután megjelenik a bejelentkező képernyő, és elolvasható a README fájl. A következő képernyőn a Licence Agreementet fogadhatjuk el, majd a sorozatszám megadása következik: ide csupa 1-es kell beírni. A következő

lapon a Visual Studio korábbi változataihoz tartozó eszközök eltávolítását kérhetjük (ez a lap csak akkor jelenik meg, ha a telepítő egyáltalán talált ilyet a gépen).

Most egy nagyon fontos rész következik: a „Visual Studio 6.0 Enterprise Edition” című oldalon az „Enterprise Setup Options” dobozból kell kiválasztani a telepítés fajtáját. Márpedig a Visual Basic 6.0-nak ez a korlátozott változata csak akkor telepíthető, ha itt a „Products” opciót (a középső lehetőséget) választjuk.

A következő lapon a Visual Studio-ban több eszköz által is használt komponensek helyét jelölhetjük ki. Ismét egy fontos lépés következik: a „Visual Studio Individual Tools Setup” című képernyőn csak a „Visual Basic 6.0 Enterprise Edition” opciót szabad kijelölni.

Ezután elindul a Visual Studio 6.0 telepítőprogramja, és a kijelölt merevlemezre másolja annak összetevőit.

Újraindítás után

A másolás befejeztével a telepítőprogram újraindítja a számítógépet. Újraindítás után ismét megjelenik az „Installation Wizard for Visual Studio 6.0 Enterprise Edition”, mert még hátra vannak a következő lépések:

Az „Install MSDN” című képernyőn ne kérje az MSDN telepítését. Az MSDN

természetesen szerves részét képezi a teljes Visual Studio 6.0-nak, de erre a CD-mellékletre nem fért fel. Tovább lépés előtt megjelenik egy párbeszédablak, amely az MSDN-telepítés elmaradására figyelmeztet bennünket a „Do you want to continue without installing MSDN?” kérdéssel. Itt válasszuk a „Yes” gombot.

A következő képernyőn („Other Client Tools”) az InstallShield telepítést választhatnánk, de helyhiány miatt ez a termék sincs rajta a CD-mellékleten, kérjük, hogy innen is lépjen tovább a „Next” gomb megnyomásával. Az utolsó előtti képernyő a „Server Setups” címet viseli, kérjük, hogy itt is lépjen tovább a „Next” gombbal. Az utolsó képernyő az esetleges regisztrációs lehetőség. Ezzel a Visual Basic 6.0 béta-változatának telepítése megtörtént.

Ismert problémák

— Az integrált fejlesztőrendszerben nincs help funkció. Ez a hiányosság a Visual Studio nagyfokú integráltságának következménye, ugyanis az összes fejlesztőeszköz közös súgója az MSDN-rendszerben található.

— A File – New Project menüpont hatására megjelenő párbeszédablakból az „IIS Application” kiválasztása hibát okoz, még akkor is, ha a hibaüzenet által hiányolt Personal Web Server vagy az IIS 3.0 egyébként megtalálható a gépen. Ez a probléma azonban megkerülhető úgy, hogy a .Program Files\Visual Studio\VB98\TemplateProjects\IIS Application.vbp fájlt közvetlenül nyitjuk meg.

Simay Endre István



A katedrális és a bazár

Miért éppen a Linux?

A statisztikák szerint jelenleg mintegy 5-6 millió ember használ Linuxot, ami ugyan lehet, hogy jól hangzik, de a számítógépen dolgozó százmilliókhoz képest nem valami sok. Tehát a Linux fontossága semmiképpen sem felhasználóinak számából következik. Akkor már sokkal inkább abból az egyes hozzáértők által is vallott meggyőződésből, miszerint ez a Unix-szerű, lényegében ingyenes operációs rendszer ugyanolyan léptékű áttörés előtt áll, mint amilyen előtt az Internet állt a '90-es évek elején, és a PC-világban pár éven belül akár a második helyre is felzárkózhat.

Eric S. Raymond, aki hosszú évek óta a Free Software Foundation és a GNU Project tagja, arról ír a Linux mibenlétét és jelentőségét (is) taglaló egyik cikkében, hogy alapvetően két-féle szoftverfejlesztési út létezik. Az egyik az ún. katedrális módszer — mindeddig leginkább ehhez vagyunk szokva. Ez a megoldás azon a feltételezésen alapul, hogy szükség van egy-fajta „kritikus komplexitásra” az igazán nagy célok és igazán bonyolult szoftverek megvalósításához, mint amilyen például a Unix világából ismert Emacs szerkesztő is. Vagy amilyen egy operációs rendszer.

Az ilyenek csak úgy alkothatók meg, ahogyan a katedrálisok is épültek, vagyis számos szakember összehangolt munkájával, és most ebből a szempontból mindegy is, hogy a Microsoft legújabb operációs rendszeréről van-e szó, vagy egy GNU projektről. A lényeg mindenképpen számos szakember összehangolt munkája.

A fenti megközelítési módnak éppen az ellentéte volna a „bazár” a maga zűrzavarával és már-már „promiszkuitásba” hajló módszereivel (ahogyan Raymond fogalmaz), vagyis azzal, hogy nincs a munkát szervező egyetlen központi hatalom. Ehelyett viszont van rá lehetőség, hogy bárki bármikor bármit megváltoztathasson, átírhatson, továbbfejleszthessen.

Utólag persze nem nehéz okosnak lenni, de valójában semmi meglepő nincs abban, hogy annak idején másokkal együtt a Free Software Foundation ezen oszlopos tagja sem jósolt valami fényes jövőt a Linuxnak. Hiszen legalábbis nehéz volt elképzelni, hogy ezzel a módszerrel egy valóban stabil és jól működő operációs rendszert lehet

létrehozni. Pedig éppen ez történt, és a Linux várható jövőbeni sikerei is éppen a bazármódszeren alapulnak.

Kezdjük az elején!

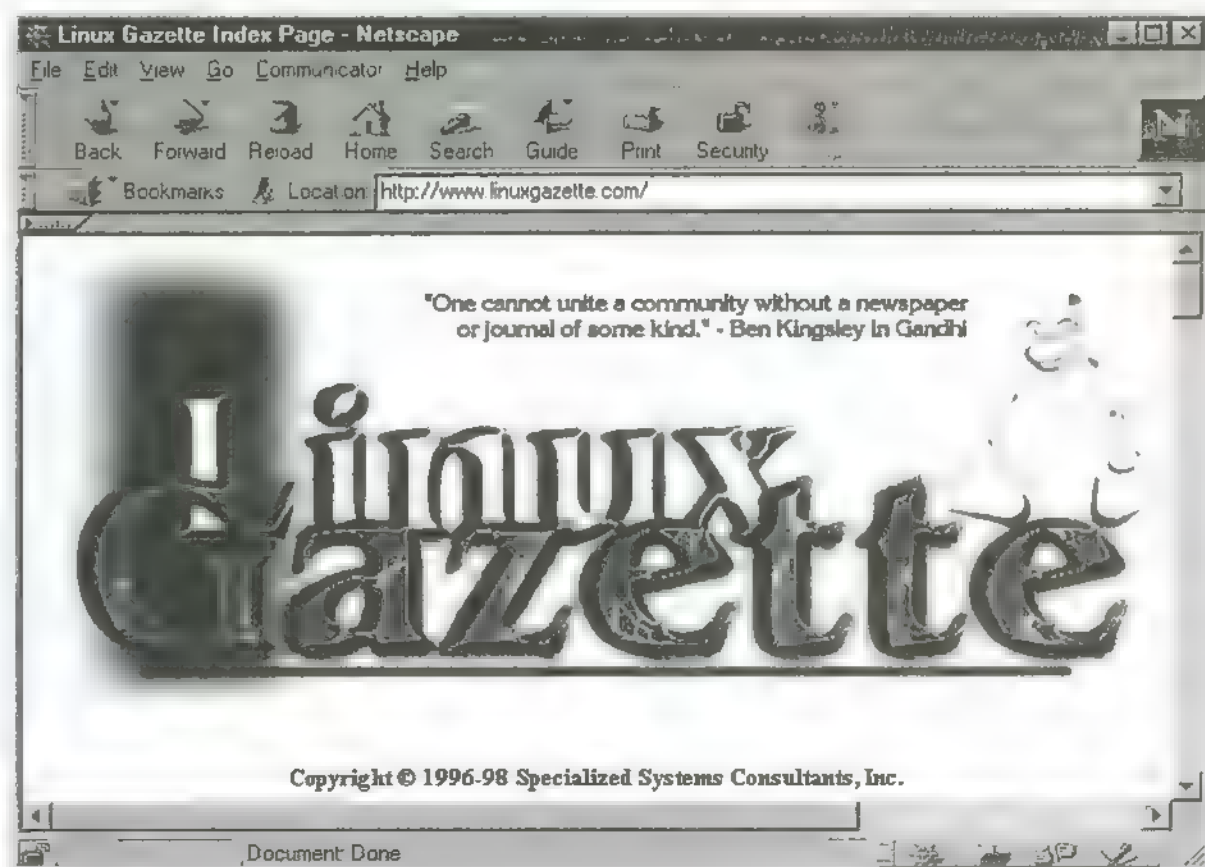
A katedrális és a bazár szembeállítását elég érzékletes ahhoz, hogy ne kelljen hosszasan magyarázni. A Linux szempontjából egy alapvető dolgot azonban ki kell emelnünk. Az a tény, hogy a Linuxra a kezdetektől fogva az „idő előtt” kibocsátott béta-verziók a jellemzőek, azért volt fontos, mert ezáltal kapcsolódhattak be mások is a munkába már a fejlesztés nagyon korai szakaszában. Ennek az összefogásnak persze volt egy másik főszereplője is: az Internet.

Az emberek csak azért kapcsolódhattak be, mert a Linux első változata 1991-ben jelent meg, amikor az Internet már nem csupán létezett, hanem viszonylag széles körben is hozzáférhető

volt. Igaz, nem százmillióknak, mint manapság, hanem legfeljebb néhány tízezer vagy százezer felhasználónak, de a földrajzilag egymástól elszigetelt számítógép-rajongók még pár évvel korábban is képtelenek lettek volna ilyen szintű összefogásra, most ennek már nem volt akadály.

Még egy fontos tényező játszott közre a dolgok kedvező alakulásában. Akkoriban az Interneten lévők leginkább fiatal, számítástechnikai érdeklődésű férfiak voltak, akik lényegében tovább vitték az eredeti (valamikor a 60-as évek elején az egyetemi központokban kialakult) számítógépes szubkultúra hagyományait. Meg voltak róla győződve, hogy a számítógép olyan dolog, ami hozzá fog járulni a világ jobbításához; és azt is biztosra vették, hogy az üzlet viszont rossz. (Vagy legalábbis nem jó. 1995-ben még egyáltalán nem számított etikus dolognak az Interneten hirdetni.) Továbbá: a mai Internet-használók túlnyomó többségével ellentétben az is természetes volt számukra, hogy tudjanak programozni.

Amikor tehát valaki egyszerre csak megjelent a színen egy szabadon fejleszthető operációs rendszerrel, akkor egyből megvoltak hozzá a közreműködők is. Valószínűleg legalább részben ebben rejlik a Linux sikerének titka: a megfelelő ötlet a megfelelő helyen. Meg persze a megfelelő időben is.



Természetesen nem ez volt az első „szabad” operációs rendszer, de Andrew Tannenbaumnak például, a hollandiai számítógépgurunak — az operációs rendszerek és a hálózat szakértőjének, aki a Minixet fejlesztette ki — eszébe sem jutott, hogy önként jelentkező kívülállók is ilyen szinten vonjon bele a munkába. És különben is: a Minixnek volt legalább két komoly gyengéje. (A sors ironiája, hogy ezeket eredetileg az erősségeinek tekintették.) Az egyik, hogy Tannenbaum professzor úgy konstruálta meg, hogy 8088-as, vagyis 1991-ben már kirívóan gyengének számító gépeken is fusson; a másik pedig az, hogy hordozhatóra, vagyis számos különböző hardveren futtathatóra csinálták meg.

Két olyan cél, ami lehetővé teszi ugyan, hogy a felsőoktatás bármely korszerűtlen komputerén is használni lehessen — cserébe viszont persze nem volt valami hatékony. Ami minden körülmények között elboldogul, az semmilyen körülmények között nem működik optimálisan. De a Minix azért ennek ellenére sem volt annyira népszerűtlen, hiszen lehetőséget adott a használóinak, hogy megismerkedjenek egy Unix-szerű operációs rendszerrel. Például 1991-ben a Helsinki Egyetem egyik hallgatója, Linus Torvalds is a Minixből indult ki, amikor sikerült hozzájutnia egy AT-386-oshoz. Linux-rajongók szerint felmérhetetlen szerencse, hogy nem egy Macintosh mellé ült le, hiszen ki tudja, hogy milyen ikonvezérelt oprendszer született volna meg... — de hát nem ez történt, Torvalds pedig elhatározta, hogy behatóbban is tanulmányozni fogja a gépe nyújtotta lehetőségeket.

Linux 0.02

„Az eredeti Linux nem volt egyéb — mondja egy, az Internetről letölthető történeti összefoglaló —, mint lényegében két feladat végrehajtására képes program, ahol az egyik feladat az AA-AA, a másik a BBBB karaktersorozat kinyomtatása.” 1991. július 3-án azonban Torvalds már azért írt a comp.os.minix nevű newsgroupba, hogy küldjék el neki az érvényes Posixot (a Posix egy általános standard, amelyhez minden Unix-verzióknak igazodnia kell), és a 0.01-es verzió már szeptemberben meg is jelent a nic.funet.fi/pub/OS/Linux alkönyvtárában — ez azonban még mindig igen távol állt a ma milliók kedvencének számító alternatív operációs rendszertől. Ugyanis nem csupán Minix kellett a futtatásához, hanem egy, ahhoz a bizonyos, Torvalds tulajdoná-

ban lévő 386-oshoz a megszólalásig hasonlító gép is. Ráadásul ekkor még maga a Linux megalkotója sem hitte volna, hogy valami nagyobb lélegzetű vállalkozásról van szó. Egy augusztus 25-i levelében ugyanis azt írta, hogy a Linux soha nem fog mást, mint IDE-harddiszket támogatni, és különben is: annyira kihasználja a 386-os adottságait, hogy soha nem lesz képes másmilyen gépeken működni...

Az 1991. október 5-én megjelent 0.02-es verzió már tartalmazott ugyan néhány binariest, de még mindig Minix kellett ezek kompilálásához, és nem volt sem floppymeghajtó, sem virtuális memória. Erre a december 11-én megjelent, és utóbb a Linux történetében mérföldkönek tekintett 0.11-es verzióig kellett várni — ami egyébként már egyből a bash shellbe bootolt fel, és ami sokkal fontosabb: már megállt a saját lábán, a Minix nélkül is. A SCSI-t viszont még mindig nem támogatta. Az 1992 áprilisában megjelent 0.95-ös viszont már minden elképzelhető eszközt támogatott, és megjelent az X Window is, hogy ezáltal végleg semmivé váljanak a Linuxszal kapcsolatos eredeti elképzelések. Mármint azok, amelyek szerint ez az operációs rendszer csak nagyon speciális feltételeknek eleget tevő gépek szűk skáláján fog futni, és ennek megfelelően igencsak korlátozott lesz a felhasználók létszáma is.

Visszanézve az is nyilvánvaló, hogy miért vált semmivé ez a feltételezés.

Raymond a szoftverfejlesztéssel kapcsolatban a „konstruktív lustaságról” beszél, értve ezen, hogy „majdnem mindig könnyebb egy jó részleges megoldásból kiindulni, mint a semmiből”. Miként az a Minixből kinövő Linux esetében is történt. Azóta talán már egyetlen sornyi kód sincs belőle a mostani verziókban, hiszen újra és újra átírták az egészet, de kiindulási pontul tökéletesen megfelelt. A Linux copy-rightja pedig már a kezdetektől megengedte, hogy bárki szabadon átalakítsa a kódot — és ennél jobb megoldás talán nincs is egy szoftver terjesztésére (amely megoldás persze gyökeresen különbözik a szoftvergyártók politikájától). Amennyiben ugyanis azt szeretném, hogy a Linux az én, eddig erre alkalmatlan komputeremen is fusson, akkor semmi mást nem kell tennem, mint úgy átírni a programokat (drive-reket stb.), hogy ennek a továbbiakban ne legyen semmi akadály (persze vannak bizonyos technikai minimumok: a gépnek például legalább 386-osnak kell lennie). Tehát a végeredmény az, hogy nem lesz olyan CD, hangkártya és

bármilyen egyéb, amihez ne lennének meg a megfelelő eszközök. Másfelől pedig ott voltak a GNU szabadon felhasználható szoftverei, ha valakinek kompilerre vagy bármilyen másra lett volna szüksége.

A talány megfejtése

Mint láthattuk, Linus Torvalds eredetileg maga sem gondolta, hogy milyen hatása lesz az egész számítógépes világra az eredetileg pusztán szórakozásból fejleszteni kezdett operációs rendszer. A siker egyik oka kézenfekvő: a szó jó értelmében vett hackerek, akiket szigorúan meg kell különböztetnünk a számítógépes rendszerekbe behatoló crackerektől, és akiknek a komputer nemcsak eszköz, hanem végcél is, most egy teljes operációs rendszert kaptak játékszerül. Nem csupán megismerhették, de a törvények megsértése nélkül át is írhatták, hogy végül egy teljes, a nagy Unixszal egyenértékű, többfelhasználós, multitasking, 32 bites operációs rendszert csináljanak belőle.

De ez önmagában nem lett volna elég ahhoz, hogy olyanok is elkezdjenek a Linuxszal foglalkozni, akik azért soha nem fognak tevőlegesen hozzájárulni a fejlesztéséhez. Ha magyarázatot keresünk, akkor ott van még a dolog pszichológiai és érzelmi oldala.

A Free Software Foundation egy web page-én az a kérdés olvasható, hogy vajon „A Microsoft-e a Nagy Sátán?”, és az a válasz, hogy szó sincs ilyesmiről. Mert bár igaz, hogy a Microsoft bojkottjára felszólító webhely is van az Interneten, a Microsoft azért végső soron nem csinál mást, mint azt, hogy mindenáron el akarja adni a termékeit — és ezt roppant ügyes marketingkonceptióra alapozva, rafináltan és hatékonyan teszi. Ugyanezt akarják a többiek is, legfeljebb kevésbé eredményesek a módszereik. A korai hackeretika szerint „minden információnak szabadnak kell lennie”, és ennek megfelelően a szoftverekért sem volna szabad pénzt kérni, de az is vitathatatlan, hogy az Apple, a Novell vagy bármely másik óriás számára ugyanúgy üzleti vállalkozásról (nem pedig a világ jobbításáról) van szó, mint a nagy ellenlábás esetében. Egyes egész nagy szoftvergyártókat lehet szeretni vagy nem szeretni, lehet rajongani a termékeikért, sőt éppenséggel utálni is lehet őket, de az érzelmek mit sem változtatnak azon a tényen, hogy az elsődleges cél a pénz. És ezen még csak csodálkozni sem érdemes: a péknek sem az a célja, hogy jobbá tegye az életemet, hanem az, hogy eladja nekem az általa sütött kenyeret.

A Linux alternatívát kínál. Részben azzal, hogy aki Linuxon dolgozik, annak nem kell a BSA-tól rettegnie (mint ahogy ma a magyar számítógép-használók 69%-ának). Részben pedig azzal (és ez is fontos szempont lehet), hogy ha szeretem a Linuxot, akkor nem egy mammutvállalat malmára hajtom a vizet. Ami „odakint”, az IRL (In Real Life) valóságában talán nem tűnik különösebben meggyőző érvnek, „idebent”, a komputer világában viszont annál többet számít.

Egy szubkultúra operációs rendszere

Aki nem kizárólag szövegszerkesztésre vagy idétlen játékok futtatására használja a számítógépet, elkezd érzelmileg is viszonyulni a használt szoftverekhez. Úgyhogy az ilyen előbbutóbb valószínűleg el fog jutni a Linuxhoz is, és ezt annál is inkább megteheti, mert a mai merevlemez méretek mellett az a pár száz MB-s partíció szinte fel sem tűnik... Immár minden feltétel adott ahhoz, hogy a Linux a második legelterjedtebb (afféle társ) operációs rendszerré nője ki magát.

És hogy miért nem az első? Nos, talán az a legfontosabb ok, hogy a rohamos terjedéssel párhuzamosan a komputert egyre többen használják kizárólag munkaeszköznek vagy játékgépnek, és aki beéri azzal, hogy egy ikonra rákattintva elindítsa valamelyik programot, az soha nem fog a Linux mellett (is) dönteni. Nem szabad ugyanis elfelejteni, hogy ez az operációs rendszer mind kialakulását, mind fejlődését tekintve számítógépes és programozói hagyományok letéteményese. Programozók fejlesztették programozóknak, vagy legalábbis olyanoknak, akik hajlandóak némi erőfeszítést tenni az operációs rendszer megismeréséért, hogy aztán cserébe egy végtelenül kifinomult, bonyolult és jól kézben tartható, sőt mi több, megbízható eszközt kapjanak (amit egyáltalán nem minden kereskedelmi forgalomban kapható operációs rendszerről lehet elmondani). Viszont X Window, grafikus konfigurálási lehetőségek meg Red Hat ide vagy oda, a Linux soha nem lesz olyan „egyszerű és intuitív”, mint bizonyos nagy fejlesztők termékei. Marad a parancssor, a config fájlok és a paraméterezés. Persze sokaknak ezt már az a tény is ellensúlyozza, hogy a Linuxot használva az ember a pozitív értelemben vett hackerizmus tradícióit továbbvivő számítógépes szubkultúrának is a részesévé válhat.

Galántai Zoltán

PPP kapcsolat Linuxszal

Fel az Internetre!

Írásommal azoknak igyekszem segítséget nyújtani, akik már rendelkeznek dial-up Internet-hozzáféréssel, és szeretnék azt Linuxszal is használni. Tisztában vagyok vele, hogy ehhez manapság már minden disztribúcióban van némi segítség (például egy pppconfig nevű program képében), mégis remélem hogy segíthetek a linuxosoknak — esetleg éppen olyanoknak, akik nem tudnak (elégg) angolul.

A szolgáltatótól kapott papírkánk birtokában először is hívjunk be valamilyen terminálemulátort, akár — horribile dictu! — Windows alól is. Nincs más dolgunk, mint modemünkkel a megadott telefonszámot tárcsáztatni, majd a „CONNECT” után megnézni, hogy a terminálszerver hogyan fogad. Ez valami ilyesmi szokott lenni:

Username: (Itt lépünk be, mintha egy BBS-re tennénk!)

Password: (A jelszót már nem szükséges begépelni, tudunk mindent, ami kell.)

Ezeket jegyezzük meg (írjuk fel), később majd kellenek. A kapcsolatot ezt követően megbonthatjuk. Ideje eltűnődni a teendőkön. Ahhoz, hogy PPP kapcsolatunk legyen, először is szükséges a PPP támogatás a kernelben. Lehetséges, hogy ehhez újra kell fordítanunk. Ha esetleg már benne van, vagy modulban készen áll, akkor használjuk azt! Emellett van még egy tömörítő modul (ez mindig modulként „fordul le”) és /lib/modules/<kernel_verzió>/net/bsd_comp.o néven található meg. Töltsük be ezt is, mert hasznos: insmod <a fenti útvonal+név>!

A kernel oldaláról készen is vagyunk, már csak fel kell építeni a kapcsolatot.

Vegyük elő újra a szolgáltatótól kapott papírunkat, és szerkesszünk bele az /etc/resolv.conf állományba, mégpedig a következő módon:

domain <A szolgáltató által adott domain>

search <Szintén a fenti domain>

nameserver <Az elsődleges névszerver IP-címe>

nameserver <A másodlagos névszerver IP-címe>

Ezeket megfelelő terminálszerverrel a Windowsok át tudják venni, de a Linux nem, ezért mindenképpen szükséges ez a művelet, hacsak nem akarunk a Világhálón IP-címek alapján tájékozódni. ;-) Ha ez kész, nekiláthatunk a kapcsolatot belőni.

A legmegfelelőbb mód erre egy kis script. Vegyük elő kedvenc szövegszerkesztőnket, és pötyögjük be a következőket (részletes magyarázat a beírandó szövegrészek után egyenként):

```
#!/bin/sh exec /usr/sbin/pppd /dev/modem 115200 crtscts defaultroute
bsdcomp 15,15 \
```

noauth connect "/usr/sbin/chat -v -f <Modemscript helye és neve>"

Figyelem, a sor végén lévő backslash (fordított törtvonal: \) azt jelzi, hogy az előtte lévő teljes szöveget egy sorba írjuk! A fenti parancs meghívja a PPP daemont, a következő paraméterezéssel:

/dev/modem

A modemet használja kommunikációs egységnek, ez egyértelmű.

115200

Ez a portsebesség bit/sec-ban mérve. Ha a soros IC a gépünkben nem 16550A típusú, akkor problémáink lehetnek (adatvesztés). Olvasóim megnyugtatósára közlöm, hogy az összes pentiumos és szinte minden on-board super I/O-val felszerelt 486-os alaplapon ez van.

crtscts

Forgalomszabályozásra a hardvert használjuk.

defaultroute

A kapcsolat létrejötte után a PPP daemon a rendszer route táblájába tesz egy bejegyzést, hogy minden ismeretlen célú csomag ezen az új interface-en keresztül hagyja el a gépet (és jut el az Internetre, ahol okos routerek már célhoz viszik).

bsdcomp 15,15

Ez az előbb említett tömörítő modult paraméterezi, a "15,15" maximális tömörítést jelent.

noauth

Szükségtelemmé teszi a túloldal (a szolgáltató) számára, hogy hitelesítse magát: Alaphelyzetben ez nem volna szükséges, de sok gyanús és lefülelhetetlen hiba forrása lehet, ha kihagyjuk.

connect " ... "

Végül ez írja elő a daemonnak, hogy milyen parancsot használjon a tárcsázáshoz, az adatcsatorna-szintű csatlakozáshoz. (Magyarán: a daemon csak a modemek „összefütyülése” után kezd dolgozni, ezen parancs végzi el a „pişik” munkát.)

Tehát a „chat” programhoz írt kis scriptnek kell a modemmel tárcsáztatnia, hogy az belépjen a szolgáltatóhoz. A fájl felépítése igen egyszerű, és ha soronként írjuk, még egyértelműbb lesz: „mit várunk” és „mit küldünk”. Ügyeljünk arra, hogy a szóközt tartalmazó kifejezéseket mindig tegyük idézőjelbe. A program természetesen a modem és a vonal foglalt állapotát is képes érzékelni, ilyenkor hibakóddal kiszáll, a tárcsázás megszakad. Ezeket a válaszokat az „ABORT” prefixummal vezethetjük be. És most példának egy minimális „chatscript”:

ABORT	BUSY
ABORT	"NO CARRIER"
ABORT	VOICE
" "	ATDT <A szolgáltató telefonszáma>
<Megjegyzés1>	<Azonosítónk>
<Megjegyzés2>	<Jelszavunk>

Rövid magyarázatot is fűzök hozzá. A tárcsázás megszakad, ha a vonal foglalt (BUSY), vagy ha közben a hívó megszakad (NO CARRIER), vagy ha modemünk emberi hangot (VOICE) érzékel. A két megjegyzés helyére azt kell beírunk, amit a cikk elején a terminálemulátorban feljegyeztünk, célszerű csak a stringek végét (például „Username:” esetén elég a „name:” is!), természetesen a felírt sorrendben! Ha ez kész, mentjük el, és adjuk meg a chat-nek a fenti (első) scriptben ennek a fájlnek a teljes útvonalát és nevét. Ugye nem is volt bonyolult?

Adorjáni Gábor

Az extra CD-n: MS SQL Server 7.0

Nagyüzemi adatbáziskezelő

Az Új Alaplap mostani számához tartozó állandó CD-melléklet mellett egy extra CD-t is kaptak olvasóink. Ezen egyetlen nagy szoftver található, az MS SQL Server 7.0 adatbáziskezelő és adatbázismenedzselő programrendszer 6 hónapon át használható béta-verziója.

Az üzleti alkalmazásokban egyre nagyobb szerepet játszanak a komplex adatbázisok, amelyekben nagyon sokféle rendeltetésű információhoz kell hozzáférni. Az adatbázisokat nemcsak létre kell hozni (vagy megvenni), hanem azokat naprakész adatokkal folyamatosan fel is kell tölteni. (Kivételt képeznek a csak lekérdezési funkcióval rendelkező passzív adattárak, mint például a CD-re kerülő menetrendek vagy telefonkönyvek.)

Az indexelés korlátai

A feladatok összetettebbé válása az adatbázisokat kezelő programokat és az adatbázisok szabványait sem hagyta érintetlenül. Hosszú ideig szinte egyeduralkodók voltak az XBase-alapú adatbáziskezelők. Elég, ha a nagy karriert befutott dBase programokra gondolunk. Ezekben az adatbázisok kezelése,

az egyes elemek nyilvántartása indexszekvenciálisan történik. Az adateléréshez az adatbázisállomány mellé egy vagy több indexállományt is fel kell építenünk, melyek nyilvántartják az adatbázis egyes rekordjainak logikai rendjét. Az indexállomány pedig együtt nő az adatbázissal, és az adatbázis elemeinek előkeresése egyre lassabban megy, mert először mindig az indexhez kell lapozni, és annak alapján megkeresni a szükséges rekordot. Ha növeljük az adatbázis méretét, és elérjük a manapság már megszokottnak számító gigabájtos vagy nem ritkán terabájtos nagyságrendet, az indexszekvenciális adatbázisok és indexállományaik fizikai tárolása és logikai kezelése egyre nehezebb.

A kisebb adatbázisok kezelésére az XBase-alapú adatbáziskezelők jelenleg is megfelelőek, de a „nagyüzemi” in-

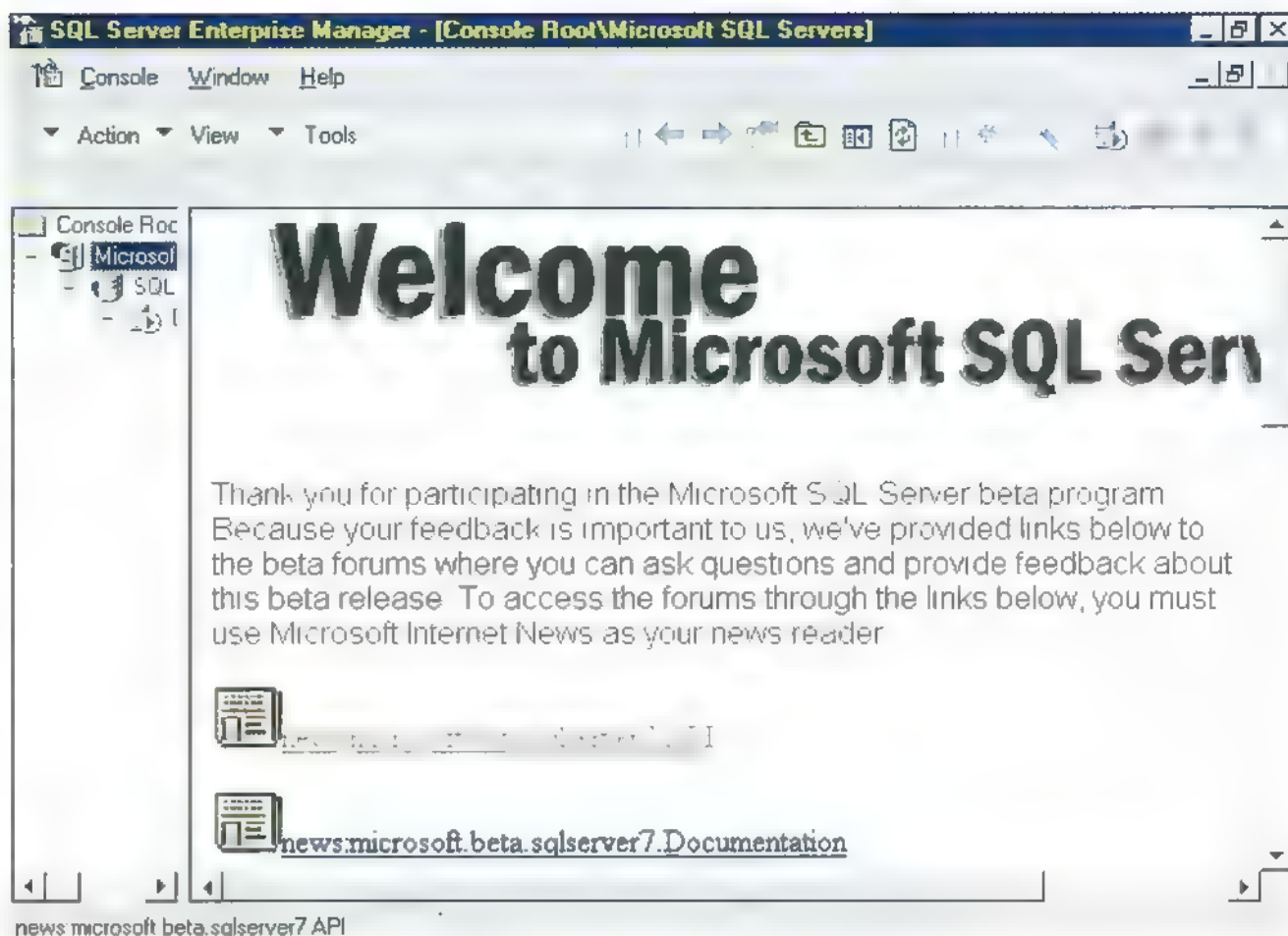


formációraktárakhoz új eszközre volt szükség. Ilyen az SQL-is, melynek fejlesztési célja kezdetben kifejezetten a nagygépes környezetben végzett munka megkönnyítése volt. Később azonban a kisebb rendszereken is megjelentek a nagyméretű adatbázisok, és azok kiszolgálásakor a központi egységekkel szembeni követelmény gyakran hasonló, mint amit a régebbi nagygépes rendszerekkel szemben támasztottak.

Ezek a PC-s rendszereken, Windows NT serveres hálózatokon végzett munkához fejlesztette ki a Microsoft az SQL Server programot, melynek korábbi, 6.5-ös verzióját hamarosan a 7.0 váltja fel. Jelenleg az új verzió béta-tesztelése folyik. Extra CD-mellékletünkre is a program 180 napig használható béta-verziója került fel. Ez egy kipróbálásra szánt, teljes értékű verzió, mely képes nagy adatbázisok kezelésének megoldására, akár többprocesszoros szerver tartalmazó rendszerekben is.

Telepítési tanácsok

Az SQL Server 7.0 telepítéséhez 32 bites Windows-rendszer szükséges. A telepítőprogram CD-ről automatikusan elindul. Ha viszont az automatikus indítást letiltottuk, akkor a CD gyökérkönyvtárából az autorun.exe közvetlenül is elindítható. Az ezt követően megjelenő menüképernyőn választhatjuk ki a további lépéseket. Bár itt az „Install SQL Server 7.0 Prerequisites” menüpontnál azt találjuk, hogy erre a



Windows 98 illetve Windows NT 5.0 esetében már nincs szükség, azonban ezeket az új programokat sok helyen még nem szerezték be vagy telepítették azokat, ezért az extra CD-mellékleten található olvassal.txt fájl tájékoztatást ad arról, hogy milyen segédletek kellenek hozzá. Ezek közül a Windows NT 4.0-hoz szükséges szervizcsomag magán a CD-n is megtalálható.

A program telepítésekor látszólag szükség van a szériaszámra, mert „kéri” azt, de ez a védelmi funkció a CD-n található verzióban még „nincs beélelve”, tehát a számmezőbe 0-t beírva a program minden további nélkül telepíthető. (Az olvassal.txt állományban

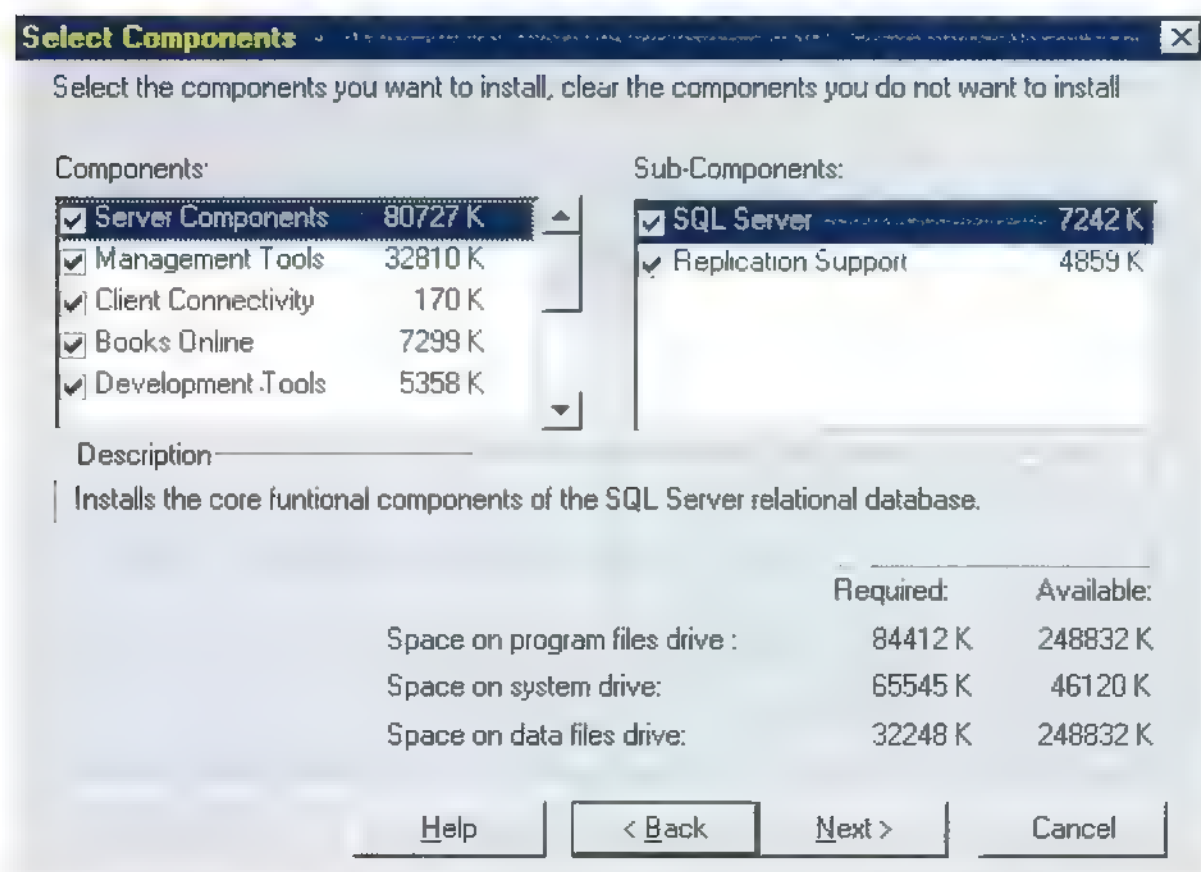
jelzett Internet Explorer 4.01 helyett a másik (a rendszeres) CD-nken megtalálható Internet Explorer 5.0 béta-verziójával is telepíthető.)

Telepítéskor javasoljuk a „custom” (egyéni) lehetőség választását, mert azal nyomon követhetjük a felrakott programrészeket. Még akkor is célszerű ez, ha az alapbeállításokon nem kívánunk változtatni, olyankor pedig feltétlenül szükséges, ha az alapbeállításban nem szereplő funkciókat is használni szeretnénk, amilyen például az SQL Server „full text search” szövegkeresési lehetősége.

A telepítést követően a CD-n is megtalálható online súgórendszert vehetjük igénybe az SQL Server 7.0 kipróbálására. Ehhez a helpen kívül több más segédanyag és dokumentáció is található a CD-n, angol és magyar nyelven egyaránt.

A program kipróbálása azoknak ajánlható, akik valóban nagy adattömeget mozgó rendszerekkel dolgoznak. A Windows-rendszereken ebben a dimenzióban jelenleg valószínűleg ez a legegyszerűbben kezelhető SQL-alapú adatbáziskezelő és adatbázismenedzselő rendszer. Segítségével közvetlen kapcsolat építhető ki eltérő belső felépítésű adatbázisok, akár még „rég” DBF-állományok között is. Kezelését pedig olyan eszközök segítik, mint a beépített automatikus konfigurálás, az SQL Profiler vagy a Graphical Query Analyzer. És az egyedi használat mellett megvalósítható a munkacsoportos fejlesztés is.

Simay Endre István



Zokszó

A „PR-cikk” szindróma

Kétévente lebonyolított felmérésünk tanúsága szerint az [Új] Alaplap olvasóinak többségét nemcsak az érdekli, ami „saját” (alap)lapjában szoftverről meg hardverről, programozásról meg alkalmazásról, informatikáról meg kommunikációról olvasható, hanem e széles szakterület sok egyéb „kelléke” is. Például a számítástechnikai sajtó, s benne lapunk helyzete. Ebből nem az következik, hogy traktáljuk olvasóinkat a lapkészítés (tényleg csak szűk kört érintő) problémáival, de az igen, hogy a mindenkire tartozó fontosabb lapügyekben folytassunk párbeszédet a lap nyilvánossága előtt. Kár lenne lemondani erről a közös gondolkodásról, különösen a mi lapunknál, amelyhez az olvasók sok szálon, nagyon aktívan és igen hűségesen kötődnek. Ezért is leheltünk új életet a Visszacsatolás rovatba, ahol ezentúl mindig közreadunk néhány olvasói véleményt, és reagálunk is azokra. Az alábbi cikk szintén része e párbeszédnek, egy „külső” esemény apropóján.

Tágabb értelemben vett házunk tájának, a számítástechnikai sajtónak a helyzetéről folyt élénk eszmecsere 1998. május 28-án az Informatikai Érdekegyeztető Fórum és az Informatikai Vállalkozások Szövetsége szervezésében, meglepően népes hallgatóság előtt. A szerteágazó, többórás vitából egyetlen gondolatfonalat ragadunk ki, amely „véletlenül” kapcsolódik Varga János főszerkesztő-helyettesnek a CW-Számítástechnikához történő átigazolása kapcsán ez év júniusi számunkban felvetett gondolatokhoz.

Anélkül, hogy bárkinek szándékában lett volna a fenti vitát a „PR-cikk” téma irányába terelni, végül mégis ez lett a hozzászólások egyik leggyakrabban visszatérő motívuma. Talán éppen azért, mert napjainkban a számítástechnikai sajtó egyre nehezebben tudja megőrizni tárgyilagos és független pozícióját. Pedig a sajtónak voltaképpen csak a médiapiac egyetlen szereplőjétől, az olvasótól volna szabad függnie. Ezt azonban különböző erők mindig igyekeznek „felülírni”.

Csak egy álarc

Az előző rendszerben a „feudálszocialista” hatalom nyíltan hirdette, hogy intézményrendszerén keresztül kézben akarja tartani a sajtó működését — persze „össztársadalmi érdekekből”. Az utóbbi 10 év alatt kialakult struktúrában viszont már a piaci automatizmusnak kellene a közjó irányába terelnie a kiadók, szerkesztők, újságírók munkáját is. Leegyszerűsítve egy kicsit a képletet: ami hasznos és szükséges a társadalomnak, annak a piaci versenyben előbb-utóbb felül kell kerekednie, ami iránt pedig nincs kellő érdeklődés és kereslet, az majd csődbe megy, átadja a helyét a hatékonyabb

megoldásoknak. A jó tehát elnyeri jutalmát, a rossz pedig az ő büntetését. Mint a mesében.

Leszűkítve a képletet az informatikai szaksajtó helyzetére, azok a lapok, amelyeket sokan megvesznek, nyilvánvalóan jobbak, értékesebbek, tartalmasabbak, hasznosabbak azoknál, amelyeket kevesebben „fogyasztanak”. Ezt az ítéletet tehát az olvasó mondja ki, tőle függ a példányszám. A számítástechnikai lapok többsége azonban kizárólag a lap eladásából (tehát az olvasóból) nem tudna megélni, ezért kialakul a függősége a hirdetőktől is. És e két függőség merőben más jellegű. Az olvasó ugyanis vagy vásárol, vagy nem, vagy előfizet a lapra, vagy nem, de döntését nem teszi függővé egyéb feltételektől. Nem mondja, például az újságárusnak, hogy „megveszem a lapot, ha kipucolod a cipőmet”.

A reklám akkor korrekt üzlet, ha mindkét fél megtalálja benne a számításait. Csak sajnos a reklám hatása sokkal áttételesebb, bonyolultabb folyamat, semhogy egy-két mutatóval, például eladási statisztikával mérni lehetne. (Az ilyen esetek inkább kivételnek számítanak.) Minden média eljut az emberek többé-kevésbé jól körülhatárolható rétegéhez, és akinek üzenete, mondanivalója, befolyásolási szándéka van ezen közönség irányába, az vásárolhat hirdetési felületet, hogy saját termékéről és szolgáltatásáról közölhesse mindazt, amit jónak, hatásosnak tart. Ha egy cég bármilyen racionális okból nem törekszik publicitásra valamilyik médiában, például mert az nem illik bele kommunikációs stratégiájába, ezt a magatartást messzemenően tisztelni kell. Talán még az is elfogadható, ha valaki nem racionálisan ugyan, de mégiscsak

következetesen teszi ezt. Annál nehezebb zokszó nélkül tudomásul venni a szaksajtóban is rohamosan terjedő, tudathasadásos „PR-cikk” szindrómát.

A PR (public relations) lényegének meghamisítását, sőt kigúnyolását jelenti, ha egy cég azzal a feltétellel hirdet, hogy jelenjen meg róla ún. „PR-cikk” is a lapban. Mondjuk ki: ez közönséges árukapcsolás, zsarolás. Már maga a „PR-cikk” elnevezés is atrocitás, mert azzal a cég tulajdonképpen meg akarja tévesztetni az olvasót. Olyan látszatot igyekszik keltetni, mintha tárgyilagos szakmai cikk jelenne meg, mintha egy független újságíró vagy maga a szerkesztőség állna a leírt mondatok mögött. Ha pedig ezt még a hirdetésektől egisztenciálisan függő szakmai lapok zsarolására is felhasználják, ha magát a hirdetést ettől teszik függővé... Szóval ennek a PR-hez már végképp semmi köze.

Nehogy félreértse valaki! A szakmai sajtóban, így a számítástechnikai publikációkban is teljesen helyénvalóak azok a cikkek, amelyek a termékekről szólnak, ismertetik az olvasóval az újdonságokat, bemutatják egy szoftvernek vagy hardvernek a működését stb. Ráadásul a független szakírók ideje is véges, és ismertetéseik nem mindig jutnak sokkal messzebbre annál, amit a cégek saját információs anyagaikban közreadtak. Azt sem kell feltételezni, hogy amikor megjelenik valamiről egy cikk, és ugyanabban (vagy egy másik) számban a gyártó vagy a kereskedő hirdet is ezt a terméket, akkor okvetlenül árukapcsolás történt. A kettő etikusan is egymás mellé kerülhet.

A szaksajtónak az a dolga, hogy kiválassza, feldolgozza és saját megközelítésében közreadja mindazt, amit érdekesnek talál, akár hirdetik azt, akár nem. A hirdetők pedig mérlegelhetik, hogy a szerkesztőség által „minden kényszer nélkül” felkeltett érdeklődést kihasználva megjelentessnek-e hirdetést is. Ha ez a kombináció morálisan nem elmarasztalható módon jön létre, akkor nincs is baj. A felszín alapján azonban többnyire eldönthetetlen, hogy miként történt valójában. (A mélység meg hallgat.)

Legalább ilyen ártalmas és morálisan elítélendő a zsarolási manőver fordított irányban, amikor maga a kiadó vagy a szerkesztőség képviselője lép fel úgy, hogy „írunk rólatok, ha hirdettek”. A szakmai lapok ráadásul elveszíthetik tárgyilagosságukat, kritikus alapállásukat, ha belemennek ebbe a zsákutcába. Fenyegető veszély, hogy fennmaradásuk érdekében feladják az olvasók tájékoztatása

iránti elkötelezettséget, mert anyagi erőforrásokat inkább remélhetnek a cégérdek kizsgálásától, a „PR-cikkektől”. És itt időzzünk el egy kicsit.

A „PR-cikk” tulajdonképpen fából vas-
karika. Matematikailag: „üres halmaz”. A „PR-anyag” egészen más. És ez nem felszínes szójáték, hanem ebben van a lényeg. PR-anyagokat bármelyik cég készíthet magáról, akár annyit is, hogy azok faxpapíron körülérnek a Földet, vagy e-mailen megtöltenék a kétgigas winchestert (ahol nem törlik le hetenként). Ha a PR-anyag eljut egy laphoz, ott kétféle formát ölthet: cikké vagy hirdetéssé változhat. Cikként formailag és tartalmilag is a szerkesztőség vállalja érte a felelősséget, az olvasó pedig ennek megfelelő követelményeket támasztva olvassa azt. A másik esetben formailag is egyértelműen hirdetésként kell azt megkomponálni, és tartalmáért teljesen a „feladó felel”. Ha egy PR-anyagból a lapban a szerkesztőségi részhez tartozó cikk lesz, akkor az információ forrásával nem is szabad a közlést megfizettetni. Ha viszont a PR-anyag alapján hirdetés készül, azt a megfelelő hirdetési tartifa szerint le kell szám-
lázni a megrendelőnek. Ennyi az egész.

A válogatás is üzenet

A fenti normák látszólag csak a fogyasztót védik a megtévesztéssel szemben, valójában azonban a cégek érdekét is szolgálják, mert enélkül a versenysemlegesség, az esélyegyenlőség a piacon nem érvényesülhetne. Ugyanakkor kialakultak a közönség önvédelmi reflexei is. Hiába próbálkoznak világszerte mindig újabb manőverekkel, tipográfiai és egyéb trükkökkel, hosszabb távon ezek nem válnak be, mert az olvasó átlát a szitán — és még dühös is lesz, hogy megint megpróbálják őt átverni. A public relations pedig ennek éppen az ellenkezőjéről, a bizalomról, a jó kapcsolatokról szól.

Mi a helyzet a cégek nagyon is újság-
vagy folyóiratszerű kiadványaival? Azokban a cikkek miért nem nevezhetők „PR-cikkekként”? Az ilyen anyagok formailag tényleg cikként, interjúként stb. jelennek meg, de van egy döntő különbség. Az olvasó egyáltalán nincs tévedésben afelől, hogy kinek az információit, kinek a véleményét olvassa, ki a lap gazdája, milyen érdeket fejez ki stb.

Az érdekeltségi viszonyokat egyértelműen, „dedikáltan” kifejező kiadvány tartalma nyugodtan lehet egyoldalú, részrehajló, elfogult... Az olvasó, kézbe véve az ilyen kiadványt, automatikusan elfogadja annak jellegét (PR-funkcióját), és eleve úgy olvassa. Azért nincs viszont értelme a benne lévő írásokat PR-cikkekként nevezni, mert a „PR-cikk” elnevezés éppen a környezetében lévő „normális” cikkektől való megkülönböztetést akarja kifejezni. Itt pedig csak PR-anyagok, vagy mellettük PR (értsd üzleti) célokat szolgáló szakcikkek vannak.

Az, hogy valami PR-anyag, vagy PR-
újság, önmagában nem fejez ki érték-

ítéletet, színvonalat. Különösen a mi szak-
területünkön gyakori, hogy a cégek házi kiadványai igen értékes és korrekt szak-
anyagokat tartalmaznak. Az újságíró, a szerkesztő az „üzemi lapoknál” is szűrő-
ként működik, de ez a funkciója fokozot-
tan érvényesül a semleges szakmai kiad-
ványok tartalmának összeállításakor. A lap terjedelmét sokszorosan meghaladó információtömegből kell kiválasztania azt, ami megítélése szerint lényeges, és ami az adott olvasótábor érdeklődését felkeltheti. A végeredmény természetesen tükrözi a válogatási szempontokat, benne még szubjektivitást is, de éppen ezekből rajzolódik ki a lapok egyénisége, profilja. Önmagát minden lapnak vállalnia kell.

Aki ismeri a public relations szakiro-
dalmát, az tudja, hogy annak egyik tömör megfogalmazása — és az erről szóló egyik híres könyv címe is — úgy hang-
zott, hogy „Tégy jót és beszélj róla!”. A cégek gyakran elfelejtik e mondat első felét (meg a PR-ből az R-betűt), és a minél nagyobb publicitás reményében inkább PR-ügynökségeket bíznak meg a munká-
val, igen kemény pénzért. Olykor erre az „ingyenreklámra” többet költenek, mint a valódi hirdetésekre. Nem csoda tehát, hogy az ügynökségek formális tel-
jesítménykényszerbe és komikus szerep-
tévesztésbe esnek, nem ritkán az értelmes kommunikációnak egyszerre több alap-
elvét is megsértve. Például ugyanazt az anyagot küldik ki a különböző profilú, illetve heti és havi megjelenésű lapoknak; nem kalkulálják be a nyomdai átfutási időt; olyanokra bízzák a számítástechnikai lapokkal való kapcsolattartást, akiknek kezében — kérdéseikből ítélve — talán soha, egyetlen számítástechnikai lap nem volt (kinyitva). De mindennek a csúcsa, amikor a PR-ügynökségek udvarias körtelefonokkal számon kérik a szer-
kesztőségeknek elküldött anyagok megjelenését, néha még azt is, hogy a cég elköltözéséről szóló hír mikor fog meg-
jelenni. (Majd ha hirdetik.)

PR ⊗ 'Yes' — Reklám ⊘ 'No'

A már említett júniusi cikkben (28. oldal) utaltunk olyan cégekre is, amelyek furcsa skizofréniával viszonyulnak lapunkhoz. Más lapokhoz is, de az ellentmondás talán nálunk a legabszurdabb, így a többiek tapasztalatát is ötvözve a mi példánk tulajdonképpen modell értékű.

Közvetlenül vagy PR-ügynökségén keresztül nagyon sok cég lát el bennünket a rájuk vonatkozó információkkal, PR-anyagokkal. Szemmel láthatóan fontosnak tartják, hogy híreik megjelenjenek lapunkban, hogy elmenjünk konferenciáikra... Nyugodtan támadhat bennünk az a kellemes érzés, hogy ők is úgy látják, ahogy mi: az Új Alaplap az a média, amelyen keresztül hatékonyan tudják informálni a számukra fontos szakmai rétegeket. Ezt cikkeink, a bennük előforduló hivatkozások még inkább felerősítik. Azt hihetnénk tehát, hogy minden a legnagyobb rendben van. Meglepetés csak ak-

kor ér bennünket, amikor ugyanezen cégek némelyike elindít egy nagyszabású reklámkampányt, és abból „kifelejtik” az Új Alaplapot, bár a hirdetések tartalma alapján azokkal kifejezetten a mi olvasótáborunkat kellett volna megcélozniuk.

Ilyenkor tehát valaki nyilvánvalóan téved. Vagy a PR-es, aki a szerkesztőséghez hasonlóan a cég célcsoportjának tekintette az Új Alaplap olvasóit, vagy a reklámos, aki nem így gondolta. Nézzük meg mindkét eshetőséget.

„No win game”

1. Ha az Új Alaplap nem tartozik a cég célcsoportjába, akkor számukra merő pénzpocsékolás szerkesztőségünk folyamatos ellátása PR-anyagokkal, munkatársaink vendégül látása konferenciáikon, sajtótájékoztatóikon stb. Mi meg pazaroljuk a faxpapírt, vesztegetjük az időnket a rájuk vonatkozó anyagok elolvasásával, megszerkesztésével... Végül azután még az olvasó is bosszankodik, hogy olyasmiről írunk, ami őt egyáltalán nem érdekli. „Ennyi hülyét egy rakáson” — mondhatná népi bölcsességgel a külső szemlélő, hiszen ezen mindenki csak veszít — kivéve a PR-ügynökséget. Csak a reklámbüdzsé gazdái vannak észnél, ők tudják, hogy nem ez a megfelelő célcsoport, és nem is hirdetnek. De akkor miért nem szólnak legalább saját PR-es kollégáiknak, hogy ne költsenek már értelmetlenül a közös kasszából!? Rejtély!

2. A másik lehetőség, hogy az Új Alaplap olvasótáborában valóban a cég által elérendő célközönséghez tartozik, és a PR-esek szakszerűen és jól dolgoznak. És akkor a reklámosok? Nem kellene esetleg közölni velük, hogy a PR-re fordított összeg az ő áldásos közreműködésük révén teljes egészében kidobott pénzzé válik? Hogy ez azért túlzás? Inkább ellenkezőleg! Sokkal nagyobb kárt okoznak cégüknek. Mert mi történik ilyenkor? A PR-munka eredményeként folyamatosan megjelenő szakmai cikkek, hírek és egyéb anyagok kialakítják a lap olvasóinak egy részében az igényt a cég termékeinek beszerzése és használata iránt, de a vásárlásra ösztönző vagy ahhoz támpontokat adó végső reklámpulzus elmarad. Pontosabban a kereslet kielégítése áttér-
lődik a konkurenciához, amely ugyanazt vagy a helyettesítő terméket hirdeti az Új Alaplapban. Márpedig az ügyfél kez-
ügyében lévő hirdetésnek nagyobb esélye van arra, hogy befolyásolja a végső döntést. A PR-esek így szépen kikaparják a gesztenyét a konkurens cégnek.

Ha ez a folyamat így folytatódik, akkor „PR-cikkekből áll majd az egész világ”, aminek a „PR-cikkek” gazdái eleinte még örülni is fognak, amíg észre nem veszik, hogy abban a kommunikációs közegben az emberek már azt sem hiszik el, ha valaki színtiszta igazat mond. És akkor majd ismét keresni kezdik a (még megmaradt) kritikus és független médiákat, melyek az olvasó számára hitelesek.

Faklen Pál

Kritikai vitriológia

Szomorú szenzációk

1998 júniusának legfontosabb híre: az előre jelzett időpontban megjelent a Windows 98, és ráadásul a Microsoft — köszöni szépen — szintén jól és még egyben van, hiába támadt neki az USA igazságügyi minisztériuma, mint fővádoló, miközben a válla felett kárörvendően liheg a Sun és a Netscape.

Előbb azonban egy kicsit politizálni fogok.

Ne keressenek összefüggést a választások utáni kormányváltás és az én témaválasztásom között, nem ÉN időzítettem így. Találgathatnak, hogy ki vagy kik.

Albacomp „inside”

Hadd vonjak egy merész párhuzamot ott a Microsoft meghurcolása és itthon az Albacomp tájkán támadt mesterséges zavarok között. A hazai számítástechnika legnagyobb és egyben legerősebb (ráadásul folyamatosan fizetőképess) cégének három vezetőjét orvul törbe csalták és bezsupolták (vizsgálati fogság). Az ellenük felhozott vádak nem tűnnek meggyőzőnek: jogosulatlan áfavisszaigénylés, fiktív számlák, esetleg csempészet és hitelkártyák hamisítása. Ez utóbbival nem is az Albacomp vezetőit gyanúsítják, hanem egy vélelmezett szolnoki partner fantomcég egyik emberét. A vádak így együttesen a mai napig nem is hangzottak el, nem írták le azokat. Másfél éve gyűjtik a bizonyítékokat. Akkor hol vannak? Mi a cél?

A nyomozóhatóság egy valamit nagy valószínűséggel el fog érni. Hírbe hozta a céget, és a finnyás magyarországi piac kerülni fogja az Albacompot, mintha leprás lenne, még akkor is, ha a cég vezetői kivágják magukat a csávából, és a vádak nem fogják megállni a helyüket. A Miniszterelnökség Közbeszerzési és Gazdasági Igazgatósága mindenesetre azonnal felfüggesztette a központi közbeszerzési tender keretében két kategóriában is győztes Albacomppal már megkötött szerződéseit. Mernék fogadni, hogy az Alba elleni feljelentést is nagy valószínűséggel egy korábban az Albacomp által szállított „gyatra” minőségű számítógépen szerkesztették. (A gyenge minőségre és az azzal való megtévesztésre történt célzás ugyancsak a letartóztatási akció „PR-jeként” hangzott el, ami viszont igen érdekes precedenst teremthet a hardver- és szoftvertermékek piacán... Ha netán

mégis komolyan akarják venni, további tippekkel mi korlátlan mennyiségben állunk az érdeklődők rendelkezésére, de talán az is elég, ha rendszeresen olvassák az Új Alaplapt.)

Az „Albacomp-ügy” várható következménye tehát a jóformán egyetlen VALÓDI nagy hazai gyártó lezüllesztése, kivézetése. Van elég „hardversenytárs”, akik majd betömködik az esetleg keletkező űrt. Például a kormányzati közbeszerzési tendereken. Kinek is az érdeke? Qui prodest? Maku látlan importőr céget aligha lehetne találni, kivéve talán azokat (fél napig), amelyek a mai napon kapták meg a külkereskedelmi jogot. Az Országos Vám- és Pénzügyőrség nyomozói csukott szemmel is megtalálhatják az addig lehúnyt és lesúnyt szemmel szándékosan észre nem vett, jöllehet mindenki által alkalmazott trükköket, és a törvény teljes szigorával csaphatnak majd le. Az új kormány esetleg fontolóra veheti, hogy létrehozza a Vidékfejlesztési Minisztériumot is, ahol egyetlen (liba)tollvonással rendezni lehet a kérdést. Minek is vidékre hazai szerelésű PC, meg országos szerviz... megoldják a magyarországi leányvállalattal rendelkező multik.

Nálunk mindenkit megillet a jóhírnév védelme, az ártatlanság védelme. Egyenlőek vagyunk, bár a híres orwelli idézet nem vesztette érvényét, és a demokráciákban is vannak másoknál „egyenlőbbek”. A HVG például teljes nevén nevezi az Albacomp három prominens vezetőjét, míg a velük együtt említett, lopott személyi adatokkal manipuláló N. Jánost (jaj, csak nem Neumann János) és K. Pétert jótékony anonimitásba burkolja. Erről persze nem a HVG tehet, ők is a hiányos rendőrségi közleményekből hámozták

ki az információkat. Az egyik oldalon ott van néhány székesfehérvári szakember, akik letettek már egyet s mást az ország asztalára, befizetett adóban és járulékokban, sőt munkában, hozzáértésben, országos szintű infrastruktúrában. A másik oldalon N. J. és K. P. személyiségi jogait bezzeg védeni kell. Hiszen ők csupán létrehoztak egy-két fantomcéget, banki átutalási stiklikkel sápolták felebarátaikat...

Visszatérve a minőség témájára, a Jász-Nagykun-Szolnok Megyei Rendőrfőkapitányság (jelesül Lengyel György alezredes, sajtófőnök) bedobta a köztudatba, hogy az Albacomp gépeibe rossz minőségű tajvani alkatrészeket építettek be, és úgy tüntették fel a dolgot, mintha azok Írországból származtak volna. Nem tudom, mi az igazság, de azt hallottam, hogy európai államhitelek (Phare program) csak európai származási igazolással rendelkező gépekre járnak. Egyébként pedig, ha a VPOP „szakemberei” belenéznek az úgymond márkás gépek vagy szoftverek belsejébe, és azt minőségvizsgálatnak vetnék alá, itt kő kövön, szem szárazon nem maradna.

Székesfehérvár egy reggeli rádióműsorban melléállt a letartóztatott vezetőknél. Akkor a börtön azonban nincsen. A szűkebb értelemben vett szakma hitetlenkedési nyilatkozatokat tett, de a tények pontos ismerete hiányában senki nem ment túl messze. Az általános jellegű állásfoglalásokban pedig azért olyan óvatos mindenki, mert tudja, hogy a magyar vállalkozó „gumiszabályok” között bokszol. Ez nagyon előnyös is lehet, amikor a gumikötél a nekifeszüléstől megnyúlik, de kellemetlen, amikor utána összehúzódik. Magyarországon a „ring” egyelőre még olyan, hogy abban a mérkőzések kimenetelét megannyi tudás, érték, érdem, tehetség, minőség, tisztesség sem tudja egyértelműen eldönteni. Ráadásul a pontozás is elég szubjektív műfaj. Egyszerűbb tehát annyival elintézni az egészet, hogy „ma neked, holnap nekem”.

Mint „illetékest”, a Microsoft magyarországi leányvállalatának vezetőjét, Reisz Attilát a rádió azonnal meginterjúvolta. Szegény, mit mondhatott volna mást, mint azt, hogy korábban

megkötött szerződéseik szerint szállítanak az Albacompnak. Tehetnének mást? Lehet, mert a hajlam a jelek szerint megvan bennük: partnereiket mindenféle szoftvertisztasági nyilatkozatokra akarják rábírní, miszerint X jelentse ki hivatalosan (deklarálja papíron) Y-nak, hogy tiszta. Ezt azonban Y szűrőpróbaszerűen bármikor ellenőrizheti Z (maga a BSA) segítségével, illetve ellenőrizi is, és akkor rettenetes szankciók, sajtóbeli meghurcolás, önkéntes pénzbírság, 2 heti csokoládé-megvonás, siralomház, akasztófa következnek. Mintha a hatályos jogszabályok adta keretek nem lennének elegendők.

Windows 95 + 3

Az Albacomp esete mellett szinte eltörpül a másik eget rengető nagy hír, hogy itt a Windows 98. Hurrá, vegye, vigye, aki akarja! Nem hemzseg a szembeszökő újításoktól, de a Windows 95 megjelenése óta még csak három év telt el, és ne feledjük, hogy Win95 tökéletes minőségű termék (nem pedig írországinak álcázott tajvani ócskaság), viszont ideje volt Redmondban újra megszuhogtatni a kaszát, hátha hullik egy kis pénz világszerte...

Innen kezdve már saját szuverén tapasztalataim, benyomásaim következnek a Win98-ről. Otthoni 200 MHz-es Pentium MMX processzoros, 64 MB RAM-os PC-men észleltem az alább leírtakat. Mielőtt még a Microsoft egy általam egyáltalán nem tisztelt „support” menedzsere cáfolná, hogy én 45 másodperces betöltési időket regisztráltam, mert az szerinte csak 30, én most, 2 és fél év elteltével reagálok egy korábbi esetre. Akkor, első felindultságomban csak valami gorombaságot tudtam volna válaszolni, amit a Új Alaplap szerkesztői úgyis kihúztak volna a cikkből. Tehát most nagyon higgadtan elmondom: nem a legbölcsebb dolog mérési eredményeket úgy cáfolni, hogy az eredeti mérésben nem veszünk részt, de a mérés reprodukálására sem vagyunk hajlandók.

Mindenek előtt az a megállapítás, hogy a Windows 98 gyorsabb, mint elődje, legalábbis a képernyőterítésnél nem igazolódott be számomra. Sőt! A bittérképes ábrák megjelenítési sebessége kimondottan pocsék, szemmel követhetően lassú. Szégyen. Attól sem vagyok boldog, hogy kilépéskor „A gép nyugodtan kikapcsolható” felirat pillanatok alatt megjelenik, mert akkor már majdnem mindegy 5-10 másodperc ide vagy oda. Sőt! Megkérdezném, hogy a Win95-ben mi a francot vacakolt annyit

az operációs rendszer a végső kikapcsolás előtt, ha mostanra sikerült normális időtartamúra redukálni?!? Igaz, kezeli az USB (Universal Serial Bus) típusú perifériákat, ami most legyen csak a Win98 gondja, ne az enyém, egyelőre ugyanis nincs a láthatáron ilyen csatlakozójú eszköz széles e honban. Sem a kívánságlistámon. A Windows 98 jószerevével tehát a Windows 95 OSR-2 apróbb továbbfejlesztésének mondható.

A tényleg jó dolgok azért jól elrejtve ott vannak a csomagban. Ilyen például az eredeti CD-n található Windows 98 Resource Kit. Külön kinyomtatva is megvásárolható a Microsoft Press kiadásában, mintegy 70 dollárért (ami testvérek között is legalább 15 ezer magyar pénz + 12% áfonya), szemben a korábbi 35 dollárral. Ennek utility programjai között csücsül a Microsoft Powertoys, és közülük a TweakUI most is nélkülözhetetlen. Olyan beállításokhoz enged hozzáférni, melyektől a normál Win98 megfoszt. Mondhatnám: mindenki számára kötelező tananyag. Ugyancsak jól használható egy batch nyelv a kötegelte feldolgozások automatikus levezényléséhez. Tetszett még a Resource Kit Sampler központi indítófelülete is. A GUI tehát fejlődött az elmúlt 3 évben. De mindenek csúcsa úgyis az MS Office 2000 vagy v9.0 lesz, valamikor az év vége tájékán.

Még egy pozitívum: jó dolog a Taskbaron a Quick Launch szolgáltatás, ami nem más, mint a kérdéses program miniikonjának képe, nem a jobb oldalon, hanem bárhol másutt, és egyetlen egérgattintással elindul.

Egy telhetetlen telepítés után (mindent beikszelve) a lényegét tekintve internetes programcsalád sorakozik majd a gépünkön:

— Outlook Express e-mailező és hírolvasó program, kvázi ügyfélnyilvántartó szintű házi telefonkönyvvvel.

— Microsoft NetMeeting audio-, adat- és videokonferencia szoftver. Az utóbbihoz minikamera és mikrofon szükséges.

— Personal Web Server, hogy kilép-hessünk akár a világegyetem nyilvános-sága elé is, gyengébbek kedvéért Web Publishing Wizarddal.

— Microsoft FrontPad nevű HTML-szerkesztő, amely nem más, mint a FrontPage 98 leegyszerűsítése, lebutítványa. (Persze nem így kellett volna fogalmaznom, hanem hogy a FrontPage a FrontPad felokosított változata. Így mindjárt jobban hangzik, ugye?)

— Microsoft NetShow lejátszó, ez egyelőre maga a jövő, merthogy háló-

zati multimédiás adatfolyamok közvetítésére találták ki, de ezek kis sávszélességű kommunikációs csatornákon gyakorlatilag élvezhetetlenül lassúak, azaz hanyagolhatók.

Az egy medvéről több bőr lehúzásának tipikus esete a Microsoft Windows 98 Plus!! (A második felkiáltójelet én raktam ki, az első mindig ott van, bár eddig egyetlen hasonló esetben sem tudtam rájönni, hogy miért.) Az 50 dolláros csomagot három évvel ezelőtt is piacra dobta a Microsoft, most sem lehetett róla lemondani. De ez már meglehetősen komolytalanra sikeredett. Mert ugye újabb desktop témák (mélytengeri búvárkaland, vadnyugat stb.) falják a helyet a CD-n, elindítva pedig a gép operatív tárában. Ott van a Microsoft Golf korlátozott, nem teljes változata. „Zakson”, de olyan apróságokat is ránk akarnak sózni, mint File Cleaner, Registry Cleaner. Ezeket egyben olcsóbban megkaphatja az ember egy UnInstaller (vagy CleanSweep) csomagban, és jobban jár. Punktum.

FAR-hírek

1998. július 1-jén megjelent az 1.52-es változat. Az említésre méltó újdon-ságok közül kiemelkedik az Advanced compare plugin. Amely a korábbi formális (a fájl neve, hossza, dátuma alapján történő) összehasonlításon túl bele-néz a fájlok és alkönyvtárak tartalmába is. Kellett is. Az FTP kliensbe bekerült az idegennyelvi kódtáblakezelés, és egyéb apróbb módosítások történtek. Shift+F8-ra a belső állománynézegető immáron elviseli az unicode-os fonto-kat is. (Egészen pontosan az ilyen kód-kiosztással megáldott betűkészleteket.) A beépített editor ezentúl oszloporien-tált blokk-kijelölésre és blokk-kezelés-re is képes, blokkokat tud jobbra-balra egyesével léptetgetni. Nocsak, alakul-gat egy ügyes kis programozói editor! Interakció? Igen, mégpedig a User Me-nu-ben foglaltak ezentúl felhasználó inputot is elfogadnak:

!<Title>?<Init>!

Például: grep !?Search for:?!?

!<Search in:!*.*!lc:\far\far.exe -v -

Más szerzők által kifejlesztett újabb pluginek is elérhetők a FAR/RAR hi-vatalos weblapjáról, a www.creabel.com-ról, úgymint:

ASCII tábla, számrendszerbeli átvál-tásokra alkalmas kiskalkulátor, UU-kó-doló (a dekóder rész egyelőre még nincs kész), egy desktopra linkeket kiköpő segédprogram (amely az .LNK állomá-nyokat is megérti és annak megfelelően feldolgozza).

Herczeg József

Lotus eSuite Workplace and DevPack

Java, vagy nem Java?

A címben feltett kérdés egyre több nagy cégnél hangzik el, külföldön és itthon egyaránt. A jelenlegi PC-s számítástechnikában meglepően gyorsan honosodott meg újból a kompatibilitás, a többplatformosság és a nyíltság fogalma, s mindennek egyik fő zászlósa (háta mögött olyan cégekkel, mint a Sun, IBM, Novell, Oracle és a Netscape, hogy csak a legnagyobbakat említsük) maga a Java nyelv, vagy más nézőpontból: platform. Ez utóbbi megközelítés márcsak azért is fontos, mert így érthetőbbé válik az IBM és a Lotus stratégiája, amely a Java-történet egyik fontos fejezetéhez vezetett.

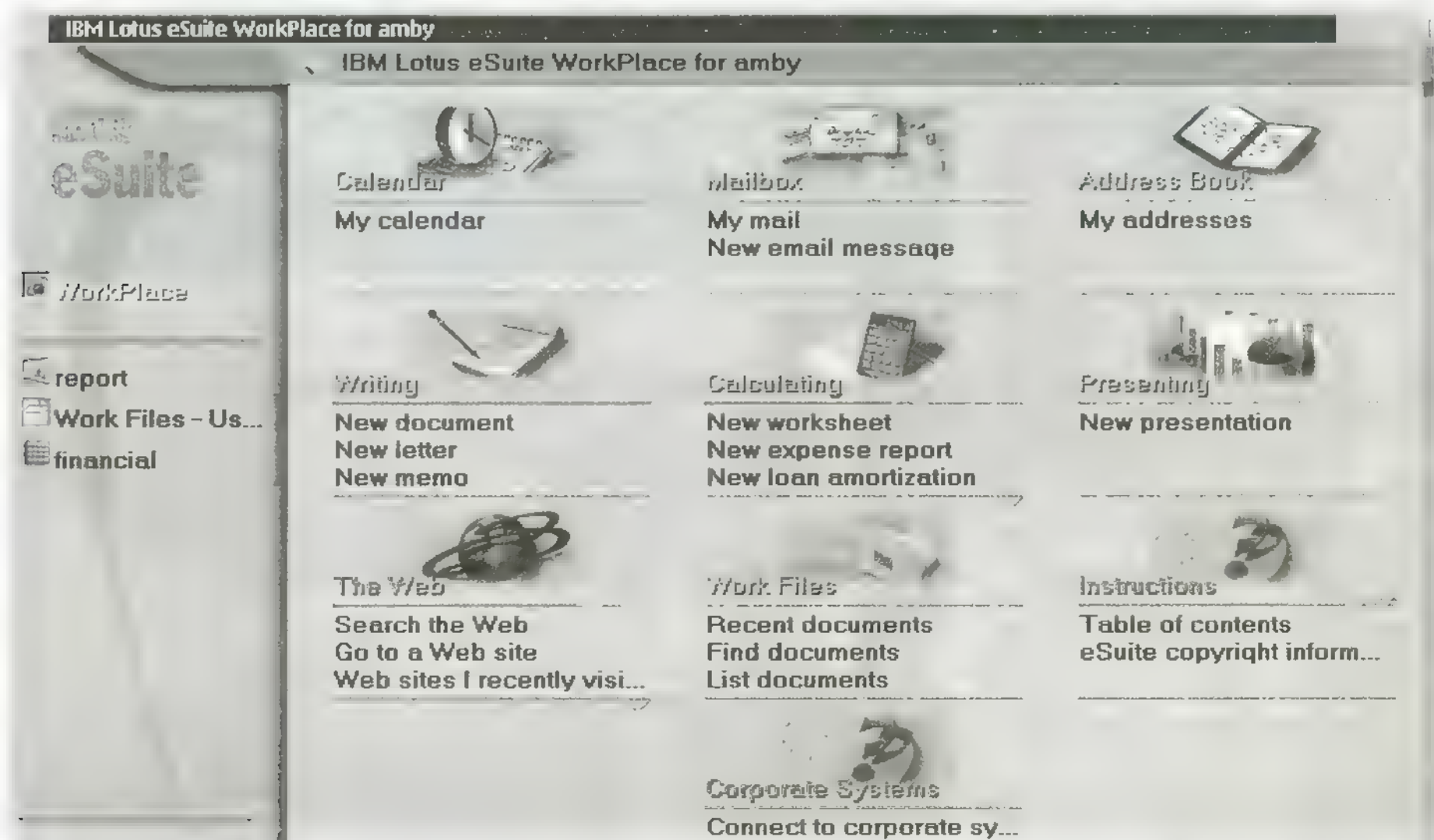
Mi táplálja a Java iránti általános lelkesedést? A számítástechnika divathullámai jönnek és mennek. Ez annak ellenére így van (volt és lesz), hogy nehéz elképzelni például a window'sos világ eltűnését. Mégis, a Java talán többet ígér (és teljesít), mind az eddigi kísérletek. Első érdekes és megkülönböztető jel, hogy rengeteg nagy cég sorakozott fel mögé, de nincs köztük a Microsoft. Másodszor: a 100% Pure Java kezdeményezésnek köszönhetően valóban egyre több alkalmazás fut szinte kivétel nélkül minden VM (virtuális gép) környezetben. A harmadik pedig, hogy rengeteg fejlesztő dolgozik Java alapú programokon, így aki egyszer

elkezd Java-ban gondolkodni, egyre több segédprogramját válthatja fel.

Magát a Java történetet is megvizsgálhatnák alaposan, most azonban az egyik legfontosabb lépést vesszük szemügyre: az első irodai csomag megjelenését Java alapokon. Elsőként a Corel próbálkozott a Corel Office for Java elkészítésével, de ez (úgy tűnik) meghaladta erejét. Fontos tény, hogy miközben a Java alapú irodai csomagokból egyelőre nem lehet sokat eladni, vagy nagy pénzeket kasszírozni, addig a Lotus-IBM párosnak a Java paradigma elterjesztése miatt pénzügyileg hosszú távon megtérülő befektetésként megéri ezen a területen alapozni.

Ennyi háttérinformáció után vegyük szemügyre magát a Lotus eSuite csomagot, amelynek két fő részéből (eSuite Workplace és eSuite DevPack) a mostani CD-mellékleten az eSuite DevPack teljes (jogilag korlátozott) változata található meg, erőforrásfájlokkal az európai nyelvekhez, köztük a magyarhoz is.

Az eSuite lényegében három különálló részből áll. A végfelhasználók inkább csak az eSuite Workplace-szel találkoznak, ez a hagyományos értelemben vett „irodai csomag”. A fejlesztők számára érdekes az eSuite DevPack, amelynek segítségével Java és Web alapú alkalmazásokat építhetnek, kész komponensekből (szövegszerkesztő, táblázatkezelő stb.). Kicsit bővebben: mindazok a Java objektumok, osztálykönyvtárak és egyebek, amelyekből az eSuite Workplace készült, részei a DevPack-nek, kiegészítve részletes dokumentációval és fejlesztési tanácsokkal. A harmadik „komponens” nem jelenik meg külön csomagban, jelentősége azonban — úgy tűnik — túlmutat az eSuite-on: ez az InfoBus technológia, amelynek segítségével a csomag komponensei egymással kommunikálnak és



amelyet külön is használhatnak a Java-fejlesztők. A Lotus—IBM és a Sun célja az, hogy szabványosítsa a technológiát, ezáltal egységesebb, dinamikus felületet biztosítson az alkalmazásoknak az adatcserére. (Távoli nézetben ez az OLE és a DDE technológiákhoz hasonló dolog.)

eSuite Workplace

A Workplace részei megfelelnek a hagyományos szerkezetnek: szövegszerkesztő, táblázatkezelő, grafikonkészítő, prezentációkészítő valamint projektkezelő és kommunikációs eszközök. Az összes modult egy közös felületről, a Workplace-ről érhetjük el. Láthatólag a tervező — mi másnak nevezhető ennek a szemet gyönyörködtető felületnek a létrehozója — nem kötődött a megszokott sémához: az erős webes hatásról árulkodó kinézet teljesen elüt a hagyományos menüs-ikonos, részben a CUA szabványból fejlődött felhasználóbosszantó egyveleg-től.

Persze ez esetben is jogos a kételkedés: hogyan valószínűsíthető meg az a funkciókavalkád, ami a mai irodai csomagokat jellemzi, ha nincs menü? Pontosan ez az, amiben az eSuite Workplace koncepciója gyökeresen különbözik elődeitől. A cél nem egy dinoszaurusz méretűre növelt komplexum, hanem egy jól használható, kizárólag a szükséges funkciókat tartalmazó, a hatékony munkát segítő csomag létrehozása volt.

A Lotus részéről a legtöbbet hangoztatott mottó a 80/20-as szabály: „Az emberek nyolcvan százaléka a funkciók húsz százalékát használja.” Az eSuite-tal elsősorban a vállalati környezeteket célozzák meg, ennek megfelelően létezik kiszolgáló része a Workplace-nek, amely elősegíti a kliensek kapcsolódását, és ott lehet beállítani mindenféle jogosultságot is a felhasználók számára, az elérhető programoktól kezdve egészen azokig a funkciókig, amelyeket az egyes modulokban használni lehet. (Módosíthat-e vlaki dokumentumokat, böngészheti-e a Webet stb.)

Ez a központilag adminisztrálható környezet teszi lehetővé, hogy a felhasználónak nem kell bajlódnia sem a felesleges funkciókkal, sem azokkal a programokkal, amelyeket nem használ. A Workplace emellett bővíthető, így bármilyen más Java program felvehető rá, helyettesítve magát az operációs rendszert és felületet. (Egy munkahelyen természetes, hogy a munkaeszközön kívül mást nem kell elérnie a felhasználónak.) Az eSuite-ot az első körben az NC-kre (hálózati számítógé-

pekre) akarják felrakni, de elkészült már más (PC-s) Java VM-ekre is a program, itt lényegében csak a tesztelés van hátra. Elméletileg minden Java VM-en futtatható a program, hiszen megfelel a 100% Pure Java kezdeményezésnek, azonban a VM-implementációk mégis különböznek egymástól annyiban, hogy csak alaposabb tesztelés után adhatók ki rájuk programok. A gyakorlatban nálunk Windows 95 alatt és OS/2 Warp 4 JDK 1.1.4 alatt is kitűnően futott.

Szöveg, táblázat, prezentáció

Itt is — mint az összes komponensben — igaz a fentiekben már kifejtett állítás: aki varázslókra, virtuális gemkapcsokra és egyéb ilyesmikre vágyik, az keressen magának másik szövegszerkesztőt... :-)

A Workplace csak az alapokat kezeli (HTML, ASCII, RTF és még néhány formátum), a DevPack programozóira vár az adott cégnél az adott környezetben szükséges extrák elkészítése. Itt nincs értelme felsorolni a funkciók listáját. Amit a HTML tud, azt lényegében meg tudja oldani a wordprocessor és a spreadsheet modul — de annál nem sokkal többet. Használhatók az elterjedt internetes képformátumok, illetve bizonyos videotípusok, melyeket a HTML is megenged.

Prezentációk készítésére a Freelance Graphics-hez erősen hasonlító modul szolgál, amely meglepően jól működött, a Java sebességének megfelelően lehetett használni.

Van még a csomagban naptár, levelezőprogram és beépített böngésző is (HotJava alapú), amelyekről nem kell bővebben értekezni, e funkcióktól el-

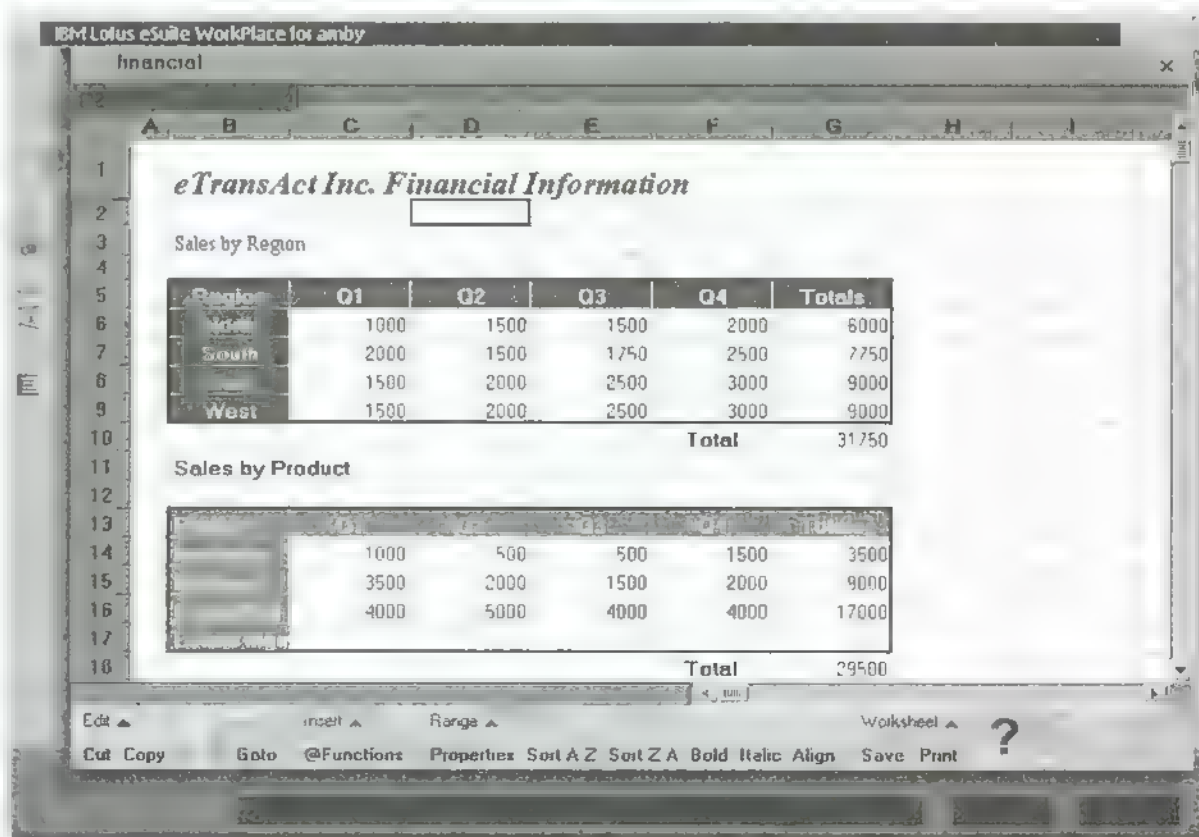
várhatóan működő, jól használható részek. Aki ismerős a Lotus Notes-ban, az a naptárt könnyen használatba veszi majd. Az eSuite csomag részei, ahol csak lehet, támogatják az elfogadott internetes, TCP/IP-s szabványokat, így például a levelezésnél a POP3-at, az SMTP-t, a címtárnál az LDAP-ot és még számos technológiát.

Összefoglalás

Ez a csomag hosszú távú koncepciójában támadás a Microsoft és más cégek által képviselt platformfüggőség ellen, ráadásul a monstrumoknak is egyértelműen hadat üzen. Leegyszerűsített készlete a legtöbb ember számára bőségesen elegendő, menüje nincs is. Ez bátor lépés, de nem is éreztem hiányát, kivéve talán az objektumokra jobb egérgombbal rákattintva előbukkanó menüket és időnként a drag'n'dropot. Úgy érzem, hogy amikor majd a program kereskedelmi forgalomba is kerül, és lefutnak az első pilot programok (hosszabb tesztelések valós környezetben), akkor sokan térnek majd át rá, főleg azok, akik egyébként is a Java felé orientálódnak. Jelenleg az IBM Network Station 1000 szériával szállítják, és ára 49 dollár!

Akire mindez esetleg még nem hatott eléggé, annak is érdemes felfigyelnie valamire: ez az első Java alapú alkalmazás, amely a végfelhasználói piacra is be tudna. Eddig a Java inkább a vállalati szférában terjedt, és a Corel-Office után kétséges volt, hogy képes lesz-e valaha is túllépni az üzleti alkalmazások körén. A Lotus eSuite Workplace és a DevPack olyan válasz, amelyre érdemes odafigyelni.

Ambrózy Gábor



Rövid hírek az OS/2 világából

A Software Choice választéka

Az IBM kiadta az OS/2-es Java motor 1.1.6-os változatát, amely elődjénél még gyorsabb, még stabilabb és még kompatibilisabb lett a Sun által definiált Java szabvány révén. Képes a több processzorral felszerelt rendszerek (Warp Server SMP) teljes kihasználására is. A Warp 4-be integrált 1.0.2-es motorral ellentétben az 1.1.6 már teljesen kiküszöböli a 2000. évi dátumproblémát is. Az Unicode támogatást nélkülöző, a Software Choice-ról letölthető változat 8 MB-ot tesz ki. Az ugyanitt található teljes verzió 18 MB, és mellékelték még a fejlesztői csomagot is (Java Development Package), amely 12 MB-os. Az időközben frissített OS/2-es Netscape 2.02 (service level 8) már támogatja az új motort, amely a J Street Mailer legutolsó kiadásával és a HotJava Browserrel is tökéletesen együttműködik. A Java 1.1.6 és a legújabb Netscape mellett az IBM elkészítette az LDAP (Lightweight Directory Access Protocol) Toolkitet is, amely azonban csak a Software Choice előfizetéssel rendelkezők számára érhető el. Az LDAP gyors terjedését jelzi, hogy már megjelent az első ingyenes, egyelőre azonban még csak szöveges felülettel

rendelkező LDAP toolkit is, amelyet az alábbi URL címen lehet megtalálni: <ftp://hobbes.nmsu.edu/pub/os2/util/network/tcpip/ldaptools.zip>.

Warp 5 vagy inkább 2000?

A WarpCast hírszolgálat (<http://www.warpcast.com/>) 63-as száma, az IBM-hez közelálló, ám magát megnevezni nem kívánó forrásra hivatkozva közölte, hogy az IBM fontolgatja egy új Warp kliens még 2000 előtti kiadását, amely a már hivatalosan 1999-re bejelentett Warp Serverrel alkotna párt. Az Interneten terjedő más pletykák szerint az IBM a Warp 2000 kiadására készül, amely az összes, eddig részeiben kiadott javításokat integrálná egyetlen, a 2000. évi dátumproblémára felkészített Warp-verzióba. Az IBM eddig azonban még nem erősítette meg a híreket.

HotJava Browser 1.1.2

A Sun kiadta a HotJava böngésző 1.1.2-es változatát, amely minden, legalább 1.1-es virtuális Java gépen, így OS/2-n is futtatható. A HotJava böngésző teljesértékű Unicode 2 és HTTP 1.1 támogatást, valamint HTML 3.2 kompatibilitást nyújt. Szinte magától értődően nem jelent neki akadályt a Java

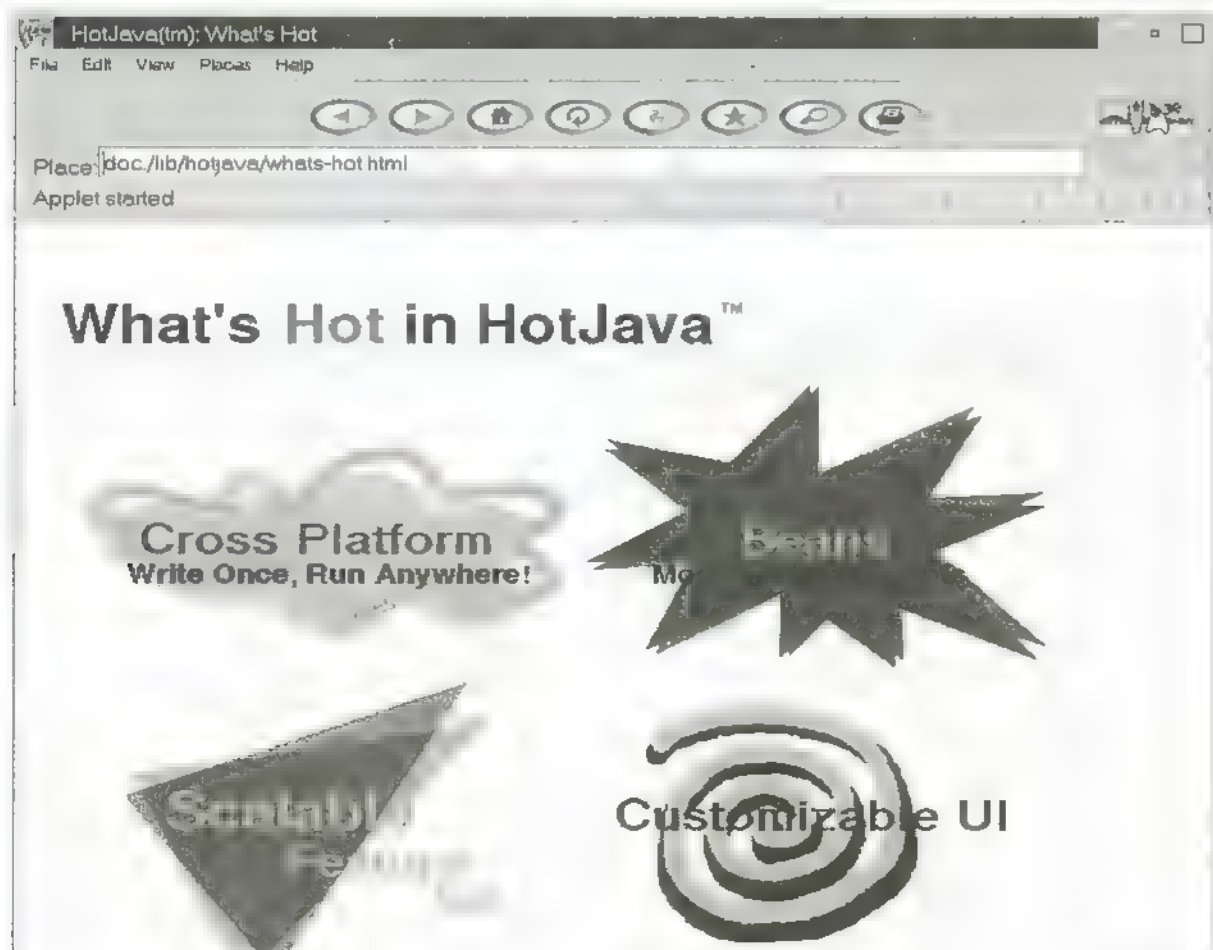
programcskákat, kereteket, táblázatokat, GIF és JPEG képeket vagy AU hangállományokat tartalmazó oldalak megjelenítése sem. A HTTP 1.1 protokoll mellett megérti az ftp és gopher „nyelvet” is, valamint támogatja az SMTP, MIME, SOCKS és SSL 3.0 (Secure Sockets Layer) szabványokat. Az alkalmazás ingyenesen letölthető a Sun oldaláról: <http://java.sun.com:80/products/hotjava/1.1.2/generic/index.html>. Akinek nem sikerülne OS/2 alatt életet lehelnie a böngészőbe, annak érdemes beszereznie a hobbes.nmsu.edu-n található hji1os2.zip csomagot, amely hasznos tanácsokat ad, és egy kiváló, REXX-ben írt indítóprogramot is tartalmaz.

MuPad 1.4.0

A német illetőségű SciFace Software GmbH elkészítette matematikai feladatok megoldására szolgáló programjának OS/2-es változatát. A MuPad már eddig is elérhető volt Windows 95-re, NT-re, Macintoshra és néhány Unix variánsra. A MuPad kiválóan alkalmas numerikus és analitikus számítások elvégzésére, illetve matematikai problémák programozására. A MuPad programozási nyelve könnyen tanulható, szintaxisa leginkább a Pascaléhoz hasonlít. A program OS/2-es verziója jelenleg nem rendelkezik grafikus felülettel, és a termék iránti érdeklődéstől függ, hogy ezt elkészítik-e hozzá. A cég jelenleg nem nyújt támogatást a MuPad OS/2-es kiadásához, viszont ingyen a felhasználók rendelkezésére bocsátja a terméket, amely az <ftp://ftp.mupad.de/MuPAD/distrib/os2/> könyvtárban található. További információk a SciFace honlapján (<http://www.sciface.com>), esetleg e-mailben is lehet érdeklődni az info@sciface.com címen.

Egyesített telefon, e-mail és fax?

Az InfoWorld (<http://www.info-world.com>) híradása szerint az IBM és a Lotus olyan forradalmi termék kifejlesztését tervezi, amelynek segítségével egyetlen csatornába lesznek terelhetők a telefon- és számítógép-hálózatokon ma még elkülönülve áramló információk. Az amerikai illetőségű Voice Technologies Group PremisMail termékét



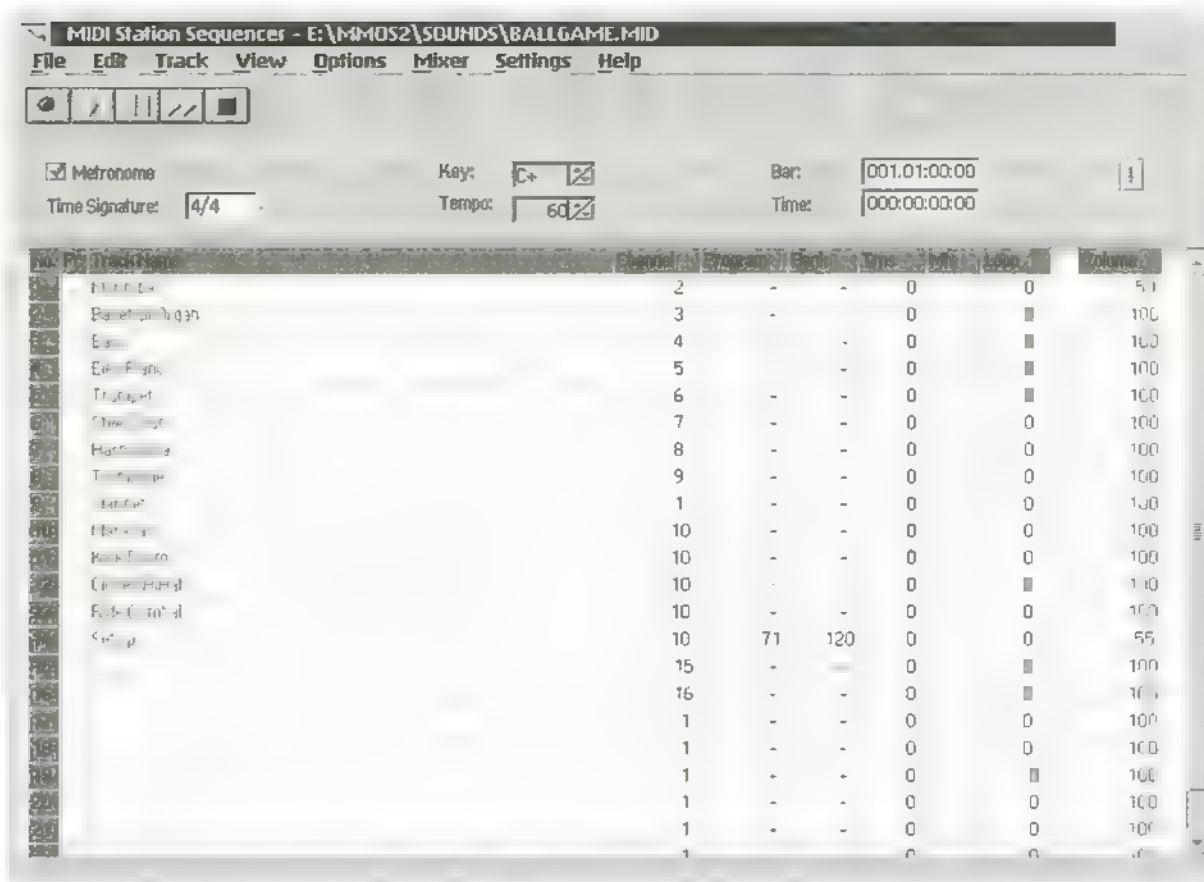
kívánják ugyanis a Lotus Notes-ba integrálni, s a tervek szerint az új Notes szerver OS/2-n futna. A PremisMail segítségével minden felhasználó kap egy ún. multimédia levelesládát, amely az e-mail-üzenetek mellett a fax- és a telefonüzeneteket is tárolja WAV formátumban, ezáltal később meghallgathatók a PC beépített hangszóróján. A program tartalmaz egy szöveg-hang átalakítót, amely biztosítja az ellenkező irányú adatcserét, hogy ezáltal az e-mail- és fax-üzenetek is kiolvashatóak legyenek a levelesládából, telefonon keresztül. A Voice Technologies Group és a PremisMail web oldala a <http://www.vtg.com> címen található.

TcpLeds 1.0

Kicsi, ugyanakkor sokak számára szórakoztató segédprogramot készített Oleg Titov. Ez a számítógép billentyűzetén található Num Lock és Scroll Lock fénykibocsátó diódákat (LED) villogtatja, amikor a gép és a hálózat között adatcsere zajlik. A TcpLeds ingyenes, és aki szeret programozással foglalkozni, annak valószínűleg külön öröm, hogy a szerző a forráskódot is mellékelte. A program egyik lehetséges lelőhelye: <http://hobbes.nmsu.edu/pub/os2/apps/internet/util/tcpleds10.zip>.

Fortify 1.2.4 a legújabb Netscape-hez

128 bites titkosítási eljárást használhatnak a Netscape OS/2-es verziójának tulajdonosai, ha Ausztráliából letöltik (<http://www.fortify.net/>) és telepítik a címben említett ingyenes terméket. Mivel az amerikai exportrendelkezések tiltják a 64 bitesnél komolyabb kódolást használó termékek kivitelét, ezért a Netscape böngészőibe beépített 128 bites titkosítási modulhoz csak az ameri-



MIDI Station Sequencer 1.06

kai, illetve kanadai állampolgárok juthatnak hozzá, ráadásul azok is csak 15 dollár lefizetése ellenében.

Warpos javítócsomagok

Az IBM „javítóműhelyéből” kikerült újabb javítócsomagokat telepíthetnek az elégedetlen vagy nyughatatlan felhasználók. A Warp 3-hoz megjelent a 36-os javító (fix), amely az ftp://service.boulder.ibm.com/ps/products/os2/fixes/v3.0warp/english-us/xr_w036 könyvtárban található. A Warp 4-hez a cikk írásának pillanatában adták ki a 7-es javítócsomagot. Ez utóbbi az ftp://service.boulder.ibm.com/ps/products/os2/fixes/v4warp/us-english/xr_m007 címről szerezhető be. Ha már a javításoknál tartunk, akkor érdemes megemlíteni az OS/2 Device Driver Pak Online

lapon (<http://service.software.ibm.com/os2ddpak/html/diskands/adaptec/index.htm>) található frissített Adaptec SCSI kártyadrivert is.

MIDI Station Sequencer 1.06

Christopher D. Hodges (cdhodge@ibm.net) jóvoltából megjelent a MIDI Station Sequencer újabb, 1.06-os kiadása. A kifejezetten Warp 4-re tervezett program egy 16 csatornás, többsávós szekvenszer, amely kihasználja a Warp 4-ben található, nagy felbontású MIDI alrendszert hangfelvételek szerkesztésére. A shareware alkalmazás letölthető a BMT Micro szerveréről (<http://www.bmtmicro.com>) vagy a szerző weblapjáról (<http://www.geocities.com/SiliconValley/Horizon/6780>).

Kádár Zsolt

DIGITÁLIS FÉNYKÉPEZŐGÉPEK

Egyszerű kezelőszervek

Memóriakártya Zoom objektív
Színes LCD Vaku



KODAK DC210

Video kimenet

KODAK
MEGAPIXEL
egymillió pontos
felbontás



NYÁRI
AKCIÓK!

KODAK

CSERÉLJEN NYOMTATÓT!

**TINTASUGARAS
NYOMTATÓJÁT
BESZÁMÍTJUK**

20.000*/10.000** forint értékben,
ha *üzemképes/**üzemképtelen

LÉZERNYOMTATÓ
79.900 HELYETT
59.900

Az árak AFA nélkül értendők



brother

HL-720
600 DPI, 6 lap/perc



Budapest, 1149 Egressy út 5.
T./f.: 221-6779, 221-6772



DIT
DIGITALTECHNIKA

Győr, 9024 Mónus I. u. 19.
T./f.: 96/414-411, Fax: 517-501

Audiatur et altera pars

„A Windows 98 az első Windows operációs rendszer, amelyet kifejezetten az otthoni felhasználók igényei alapján alakítottunk ki — nyilatkozta **Bill Gates**. — Ennek köszönhetően ez a Windows eddigi legkényelmesebb, használatát tekintve legegyszerűbb változata, amely remélhetőleg még több felhasználó számára teszi elérhetővé a PC nagyszerű lehetőségeit.”

„A Windows 98 nemcsak a felhasználók számára kínál nagyszerű új lehetőségeket. Egyedülálló jellemzői az egész PC iparág számára előnyösek — mondta **Rod Schrock**, a **Compaq** elnökhelyettese. — A hatékony, de egyszerűen kezelhető és a felhasználók igényeihez nagyszerűen alkalmazkodó jellemzők az iparág fejlődésére legalább olyan jó hatással lesznek, mint amennyire megkönnyítik majd a tapasztalt és kezdő felhasználók életét.”

„A Windows 98 USB-támogatásának köszönhető, hogy hamarosan jelentős mértékben kibővül az USB eszközök [universal serial bus] választéka, és az is, hogy az új termékek használata már sokkal egyszerűbb lesz — nyilatkozta **David Fair**, az **Intel** technológiai fejlesztési vezetője. — Az USB technológia elterjedésével egészen újfajta számítástechnikai lehetőségek tárulnak fel, és egyre többen kezdenek majd PC-ket használni, ami egyúttal azt is jelenti, hogy a piac kibővül, és a felhasználók számára a PC sokkal többet nyújt majd, miközben használata sokkal egyszerűbbé válik.”

Bankinformatika Unix-alapon

Külföldi és hazai szoftverházak versenyében, többfordulós pályázaton nyerte el a megbízást a hazai fejlesztésű szoftverek történetének legnagyobb értékű projektjére a banki számlavezető rendszerekkel foglalkozó **Online Kft.** Egységes bankinformatikai platformon kell kialakítani az ország több mint 230 takarékszövetkezetének 1700 fiókjában a számítástechnikai rendszert. A központi szerveren működő **BOSS** (Bank Operating Software System) integrált banki alkalmazás Unix alapon, Progress futtatókörnyezet segítségével szolgálja ki a helyi és távoli terminálokon bejelentkező felhasználókat.

Java tanfolyam

Magyarországon ez év tavaszán öt vezető számítástechnikai cég, az **IBM**, a **Sun**, az **Oracle**, a **Novell** és az **IQsoft** elhatározásából megalakult a **Java Szövetség**, azzal a céllal, hogy elősegítse a Java oktatását, terjesztését. A nyáron meg is rendezték az első egyhetes bentlakásos tanfolyamot (Java szemináriumot), amelyen a „Java osztályok” indítására vállalkozó tíz felsőoktatási intézmény 25 ok-

tatója díjtalanul vehetett részt. A szövetség a későbbiekben is elsősorban a tanintézetek számítástechnikát oktató tanáraihoz akar eljutni, hogy rajtuk keresztül minél szélesebb körben megismertesse és a gyakorlatban meghonosítsa a Java technológiát.

Ifabo helyett Info

Egy fejezet lezárult. 1990 decemberében kötött szerződést a **Hungexpo** és a **Wiener Messe**, hogy a bécsi Ifabo mintájára — és ugyanolyan névvel — közösen rendezzenek Budapesten informatikai szakkiállítást. Ennek szervezésére hozták létre Bécsben 50-50%-os tőkerészesedéssel az **ECI Expoconcept International** nevű céget. A budapesti Ifabo 8 éven át a hazai számítástechnika szerves része volt. A lejárófélben lévő megállapodás megújítására másfél évvel ezelőtt kezdődött az alkudozás, amelynek „eredményeként” azonban nem tudtak megegyezni a közös folytatásban. Így a **Hungexpo** 1999-ben már teljesen saját rendezvényeként, **Info '99** névvel és az Ifabónál megszokott időszakban (április 27-től 30-ig) rendezi meg az Ifabo 'de facto' folytatásának tekinthető informatikai kiállítást. Az első megnyilvánulásokból ítélve ugyanakkor az osztrákok még nem adták fel, hogy az Ifabo név tőkéjét kamatoztatva más helyszínen esetleg mégis szervezzenek Budapesten Ifabo kiállítást. Kérdés, hogy elbír-e ez a szakterület az őszi Compfair-rel együtt már három ilyen rendezvényt évente, amikor nem kevés kiállító még a kettőt is soknak tartja.

Cognos rendszerek a Synergontól

Viszonteladói megállapodást alapján a **Synergon** fogja értékesíteni az **Axis** által magyarországi disztribútorként képviselt kanadai **Cognos** cég üzleti döntéstámogató szoftvereit: az **Impromptu** végfelhasználói kimutatáskészítő, a **PowerPlay** többdimenziós elemző és a **Scenario** adatfeltáró programokat. A **Synergon** az általa forgalmazott integrált vállalatirányítási rendszereket — **Ross Renaissance 4GL**, **J.D.Edwards OneWorld** és **SAP** — a szerződés révén korszerű, könnyen kezelhető, látványos vezetői információs rendszerekkel tudja kiegészíteni.

Nyomtatógyártás Magyarországon

A **Flextronics International** bejelentette, hogy a **Hewlett-Packard** őket választotta partnerként a HP tintasugaras nyomtatók gyártására Európában. A **Flextronics** feladata a nyomtatott áramköri kártyák előállítás és a készülékek összeszerelése lesz. A gyártás a cég magyarországi telephelyén, Sárváron valósul meg. A tervek szerint az első nyomtatók 1998. szeptemberében készülnek el, és azokat a HP Európában forgalmazza. Leg-

először a HP DeskJet 720C készül itt, de később többféle modellt is fognak gyártani. A **Flextronics** termelése az európai kereslet kb. 15-20%-át fedezi majd. Európa a HP tintasugaras termékeinek világméretű forgalmából kb. 40-45%-kal részesedik, és a HP uralja az európai tintasugaras nyomtatók piacának közel 50%-át.

Infokom

Info (azaz informatika, számítástechnika) és **kom** (kommunikáció) — e két fogalom lapunk koncepciójában és szerkesztőségi anyagainak tartalmában mindig is nagyon szorosan összetartozott, csupán hirdetési palettánk volt kissé féloldalas... eddig!. Az Új Alaplap munkatársainál kevesen örülhettek hát jobban az **Infokom Egyesület** megalakulásakor rendezett sajtótájékoztató meghívójában leírtaknak. Egyrészt annak, hogy „egyre több jel mutat arra, hogy a számítástechnika és a távközlés között leomlanak a falak.” Másrészt, hogy az informatikai szakma támogatására és a közös érdekért történő eredményes fellépés céljából Magyarország 10 legnagyobb számítástechnikai és távközlési vállalatának első számú vezetője július 9-én létrehozta az **Infokom Egyesületet**. Az alapító tagok: **Ericsson**, **IBM**, **KFKI**, **Matáv**, **Nokia**, **Pannon GSM**, **Siemens**, **Synergon**, **Számalk**, **Westel 900**. Az egyesület elvileg ugyan nyitott, de a hatékonyság megőrzése érdekében csak a legnagyobb cégek csúcsvezetői előtt. Az elnöki tisztséget félévenként mindig más cég vezetője tölti be. A sort **Horváth Róbert**, az **IBM** vezérigazgatója nyitja. Az egyesület gyakorlati munkáját főállású ügyvezető igazgatóként **Niklai Péter** irányítja.

A harmadik „Oroszlánköröm”

A mobil számítógépeket forgalmazó **Portocom Rt** három évvel ezelőtt alapította a „Portocom Oroszlánköröm Díj”-at, és azóta minden évben mobil géppel, programokkal, Internet-eléréssel jutalmaznak meg egy kiválasztott, nagyon tehetséges, oroszlánkörmeit korán megmutató fiatal. 1996-ban a tévéből azóta országsszerte ismertté vált **Szász Marci** kapta meg a díjat, ő 7-8 évesen már gimnáziumi matematikapéldákat oldott meg és 198-as volt az IQ-ja. 1997-ben a választás a 12 éves **Turi Tímeára** esett, aki sokoldalú csodagyerek, verseket ír, maga illusztrálja köteteit, filmfesztiválon kapott díjat és zeneszerzési pályázatot nyert. Az idei nyertes, a 12 éves **Rákóczi Karolin** is különleges egyéniség, cigány származásának és kedvezőtlen körülményeinek hátrányait nagyon rövid idő alatt leküzdve osztályának egyik legjobb tanulója lett. Hogy mindezt elérhette, abban nagy szerepe volt a nyírteleki Kedves-ház kollégiumot létrehozó kitűnő pedagógusnak, **Lázár Péternek** is.

Stingray nevű referenciarendszereket mutatott be a Chromatic Research. A 800 dollár alatti PC gyors 3D és DVD lejátszási lehetőségét az Mpact 2 multimédia-processzor adja. Ez a chip a Pentium vagy Pentium II (és Celeron) processzort egészíti ki a 3D, a házimozi minőségű DVD, a Dolby Digital hang, a 64 hangú szintetizátor és az 56 k-s modem minden képességével — a külön összeállított, hasonló tudású megoldások töredékéért. Ez az első részletesen bemutatott terv az 1000 dollár alatti PC-k között.

>>

A Celeron 300 MHz-es változatát is bejelentette az Intel, ezzel is csökkentve a Pentium és a Pentium II processzorok közötti rést. Az olcsóbb Celeron chip-ekkel feltehetőleg a Slot 1 architektúrára való átállást kívánja gyorsítani, mert ezen a platformon jelenleg nincs konkurenciája.

>>

A Cyrix — már a National Semiconductor részeként — tovább folytatja X86-kompatibilis processzorainak fejlesztését. A most bejelentett M II chip 333 MHz-es változatban készül 0,3 mikronos technológiával, 1998 második felétől pedig átérnek a 0,25 mikronosra. A processzor továbbra is a Pentiumhoz készült Socket 7 foglalatba helyezhető. Ára ezer darabos tételben 180 dollár lesz.

>>

A Yamaha WaveForce 192XG jelentős előrelépés az ár és a teljesítmény viszonyában, mert extra hangminőséget nyújt a belépő szint árkategóriájában, vagyis 100 dollár alatt. A WaveForce (hardveres és szoftveres) szintetizátorának mindegyike 2 MB hangmintát tartalmaz, 676 hangszerrel és 21 dobkészlettel. S-VA fizikai modellező rendszere révén különösen élethű szólóhangzások állíthatók elő például rézfúvós vagy húros hangszerekből. A kártya kompatibilis a General MIDI, TG 300B (GS emuláció) és a Yamaha saját XG szabványával, és DOS-kompatibilis a Windows DOS boxán keresztül, vagy real módban a D-DMA és PC-PCI segítségével. Az S-YXG50 szoftverrel az egyszerre megszólaltatható hangok száma 128-ig terjedhet.

>>

GrabIT II néven árusítja legújabb állókép-digitalizálóját az Aims Lab. A printerportra csatlakozó kisméretű készülék egy számítógép-billentyűzethez való adapter segítségével kapja a tápfeszültséget, és képes 24 bites, akár 1600x1200-as felbontással rögzíteni a képet videolejátszóról vagy kameráról. (Ezzel a legtöbb digitális kamerát jóval túlszárnyalja.) Több előzetes ablakban hat egymásutáni képet jelenít meg egyszerre, így könnyebbé teszi a feldolgozást, és megspó-

rolja a kazetta visszatekerését. Nyomtató csatlakoztatása sem gond, mert rendelkezik printeraljzattal, nyomtatáskor tehát nem szükséges eltávolítani. (70 dolláros amerikai ára a visszatérítéses akcióban további 20 dollárral csökken, így ez a legolcsóbb grabber a piacon.)

>>

18 collos lapos képernyőt mutatott be az EIZO. Képpontmérete 0,28 mm, ezzel torzításmentes 1280x1024-es felbontásra képes, speciális interpoláció segítségével a 800x600-as és az 1024x768-as módok is közel teljes képernyős formátumúak. Az EIZO egyedi kettős bemenete lehetővé teszi egyszerre két számítógép csatlakoztatását, beállítva az elsőbbséget a portok egyikére. A megjelenítő két változatban készül, az egyik az asztali, amely USB portot tartalmaz, a másik a free-mount, azaz tetszőlegesen felszerelhető, akár falra is. Mindkét változat júliustól kapható, ára 3600 dollár.

>>

Az AccelGraphics geometriai gyorsító grafikus kártyát készített, és abba a Glint Gamma processzorát építi be. Az Accel-GMX 2000 grafikus alrendszer minden OpenGL funkciót felgyorsít, ezért nagy poligonszámú mechanikai CAD vagy animációs modellek megjelenítésére alkalmas, ahol rendkívül nagy számítási sebesség szükséges. A Gamma chip teljesítménye 2000 MFLOPS (a Pentiumhoz hasonlítva közel százszor nagyobb), a kártyán még 96 MB memória is helyet kapott, ezáltal lehetővé válik az 1920x1200-as felbontás használata teljes 3D funkciók mellett, a mintázatoknak pedig 29 MB marad. Természetesen ennek a tudásnak az ára is tekintélyes: 2900 dollár.

>>

Elsőként az ATI hoz forgalomba olyan kártyát, amelyen már van csatlakozó az új (lapos, digitális) monitorokhoz is. Az XPERT LCD a hagyományos 15 pólusú mellett 20 pontos digitális kimenettel is rendelkezik, és a kártya által alkotott képet nem kell analóg jellé alakítani, mert a lapos monitorok digitális jellel dolgoznak. Javul tehát a képminőség, mert a jelenlegi ilyen monitoroknál az oda-vissza átalakítás minőségvesztéssel jár. Ugyanakkor a monitorok ára is csökkenhet, mert nem kell átalakítót beépíteni.

>>

A Fujitsu új geometriai gyorsítója a PC-k 3D teljesítményét fokozza. A rendkívül jó ár/teljesítmény arányú Renditon V2200-as grafikus chip a FGX-1 általános célú PCI gyorsítójával együtt az IBM x86 alapú rendszereinek nyújtanak nagy teljesítményt, kivételes áron. A 200 MFLOPS feldolgozási sebességet 100 MHz-es órajelen éri el, ami másodper-

cenként akár 750 ezer poligon megjelenítését eredményezheti. A chip ára 20 dollár alatti.

>>

Hosszas várakozás után a 3Dfx nyilvánosságra hozta Banshee kódnevű, integrált 2D/3D processzorának adatait. A Voodoo2 3D képességeivel rendelkező Banshee mellé már nem szükséges normál VGA kártya, kompatibilis a DOS-szal, és a Windows valamennyi GDI funkcióját hardveresen támogatja, szemben az eddigi hibrid megoldásokkal. Emellett DVD lejátszásra is alkalmas. A külső memóriához — 4, 8 vagy 16 MB — 128 biten csatlakozik, a belső interfész 256-os. RAMDAC-ja 230/250 MHz-es, maximális felbontása 1920x1440 pixelig terjed. (Amennyiben a 2D funkciók terén is a 3D-hez hasonló képességeket lesz képes felmutatni, akkor igen népszerű kártyák épülhetnek majd rá.)

>>

Az Aopen America cég új PCI hangkártyája a Maestro-2 engine-re épül: audio-processzora 500 MIPS-szel egyenértékű, 64 csatornás hangzásra képes, támogatja a DirectX 3D pozicionáló eljárását, és az Interneten keresztüli beszélgetést több forrásból. A PCI busz nagy sebessége révén a MIDI hangmintákat a főmemóriából képes lejátszani, így nem szükséges külön memória a kártyán.

>>

Az NVidia Riva TNT processzorát egyre több cég támogatja. A Riva 128 sikerére alapozva a gyártók közül sokan bejelentették, hogy az új chipre épülő termékeket hoznak ki. Ilyen például a Diamond Multimedia, az Elsa, vagy a Canopus, amelynek előző Riva-alapú Total3D 128V kártyája egyedi fejlesztéseket is tartalmazott. Valószínűleg a Hercules lesz az első az S3 Savage3D chippel ellátott kártyák között. A Terminator Beast 8 MB SGRAM-mal lesz felszerelve, ami az S3 speciális tömörítésével akár 24 MB-nyi mintázatot képes tárolni. Képfriessítése a maximális 1600x1200-as üzemmódban is 90 Hz-ig állítható, továbbá felbontásfüggetlen videokimenet is helyet kap rajta, végfelhasználói ára mégis 200 dollár alatt marad.

>>

A Creative Labs megvásárolja a Silicon Engineeringet, egy multimédiával, társasjátékkal és kommunikációs áramkörök tervezésével, fejlesztésével foglalkozó céget. Ez stratégiai lépés egy szélesebb termékskálával történő piaci megjelenéshez. Szintén Creative hír: a cég pert nyert a Cyrix-szel szemben. Az ítélet megtiltotta a Cyrix XpertsAudio-hoz a Sound Blaster drivereinek alkalmazását és terjesztését.

Bánó György

Modellezés AutoCAD környezetben

Magyar nyelven is megjelent a Mechanical Desktop 2.0. Az új verzió az AutoCAD Release 14 — egészen pontosan az R14.01 — köré épül, annak minden újdonságával: sebességnövekedés, fotorealisztikus kirajzolás (rendering), egyszerűsített szerkesztő funkciók, kimagaslóan jó felhasználói felület stb.

Elődjéhez képest (Mechanical Desktop 1.2) legfőbb újdonsága a testmodellezésben jelenik meg. Első pillantásra is szembetűnő a modellépítést megjelenítő és azt aktívan szerkeszteni képes áttekinthető ablak, az ún. Desktop Browser. Ennek segítségével könnyen elérhetjük az egyes alaksajátosságokat, akár szerkeszteni, akár törölni vagy áthelyezni akarjuk azokat a születési fában. Segítségével gyorsan eligazodhatunk egy bonyolult szerelvényben, egy robbantott ábrában. Megjelentek olyan új alaksajátosságok is, mint például a több falvastagságú héjképzés, vagy a paraméteres Boole-algebrai műveletvégzés. A test- és felületmodellezési technológiák ebben a verzióban is ötvözhetők.

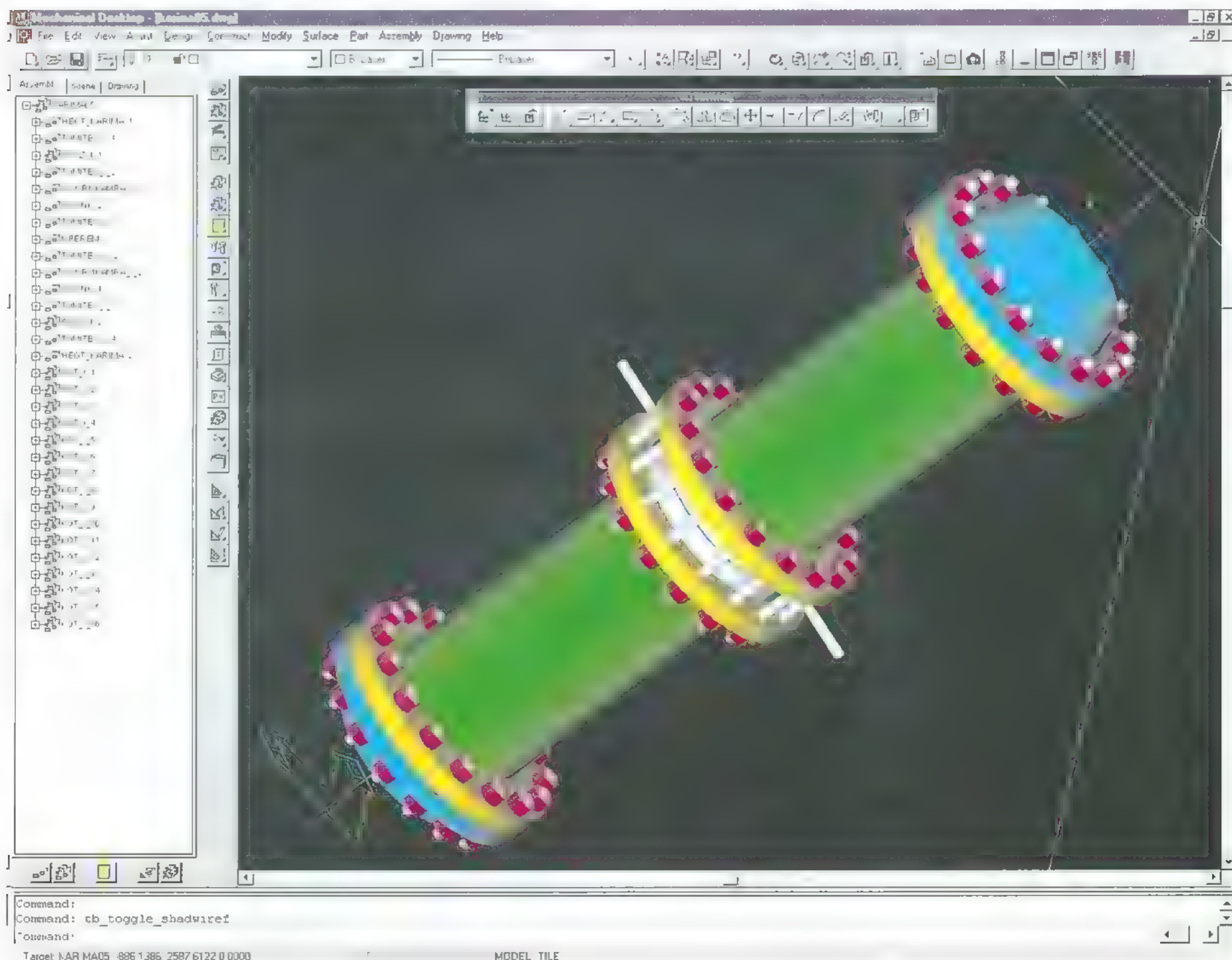
Nagyságrenddel gyorsabb lett a nézetgenerálás. Ennek titka az AutoCAD R14 és az új (Acis 3.0) modellező kernel, a modellek leírásához szükséges matematikai függvénygyűjtemény. Ezekkel az eszközökkel lehetőségünk van nemcsak egy alkatrészt, hanem egy egész alkatrészcsaládot is megtervezni. Az ősmodellhez táblázatot is kapcsolhatunk, amelyben

katalógusszerűen szerepelnek az egyes változatok méretei. A változatok között a Desktop Browser segítségével változathatunk. A modelltől készített műszaki rajzon egyszerűen helyezhetünk el helyzet- és alaktűrési, illetve felületkikészítési jeleket. A hegesztett szerkezetek jelképei szintén objektumként helyezhetők a rajzba. Ez utóbbi rajzfunkciókat külön is meg lehet vásárolni az AutoCAD-hez, a neve AutoCAD Mechanical.

Nemcsak a CorelDraw...

A Corel grafikai termékcsaládjának mindmáig legsikeresebb — és valószínűleg a legjobban sikerült — tagja a „családalapító” CorelDraw. Magyarországon is igen elterjedt, annak ellenére, hogy nyelvi problémák persze máig is vannak vele, mert magyar nyelvű szöveget görbvé nem átalakított formában tartalmazó .CDR állományok csak olyan gépeken jelennek meg ékezet helyesen, amelyeken történetesen az adott fájlhoz felhasznált betűtípusok is telepítve vannak. Ráadásul ugyanaz a típus (pl. Times New Roman) számos „vadmagyarított” fontkészlet formájában fordul elő, tehát azonos betűnév esetén sem biztos, hogy a szöveg grafikailag ugyanúgy fog kinézni a „fogadónál”, mint az „adogatónál”. És röpködnek a „kalapos” és „hullámos” ékezetek is...

A Corel sokáig nem nagyon törődött a „kis nyelvek” sirámaival, aminek egyik következménye, hogy a CorelDraw-nál még inkább szövegoreintált grafikai programjainak



magyarországi használata egészen minimális. Amikor a Corel 1993-ban megvásárolta a Ventura DTP programot, és áttette azt Windows felületre, elszalasztott egy kedvező alkalmat. Előzőleg ugyanis az SZKI által magyarított, GEM grafikus felületű Xerox Ventura Publisher Magyarországon már nagyon népszerű volt, jelentős mennyiséget adtak el belőle, ezáltal (és a nem jogosult felhasználók nagy számát sem hagyva figyelmen kívül) egy új Ventura-változat nagyon jó ismertséggel, kedvező terjesztési pozícióból indulhatott volna. Ehhez azonban hozzá kellett volna adni még egy kis honosítási munkát, amit sajnos nem tettek meg.

Az utóbbi egy évben végre elmozdult a holtpontról a Corel kelet-európai piacpolitikája. Sorra jelentek meg a számítástechnikai lapok CD-mellékletein (így az Új Alaplapban is) a Corel programok bemutatkozó változatai és 1998 májusában és júniusában egy újszerű, bár egyesek szerint vitatható amnesztiát hirdettek a CorelDraw 8-asra (kedvezményes átlépést „könyvről szoftverre”). A legfrissebb fejlemény, hogy Péntes Zsuzsa ügyvezetővel az élén megalakult a Codra Kft, a Corel termékek kizárólagos magyarországi behozatalára és forgalmazására, illetve az ezzel kapcsolatos oktatási, tájékoztató és promóciós tevékenységek szervezésére. (Ugyanakkor a Corel kötött egy másik szerződést is Magyarországon a Sved Rt-vel, bizonyos — általában nem a legfrissebb — Corel szoftverek OEM jellegű, azaz a hardverre telepített, azzal együtt történő értékesítésére, amit múlt havi számunkban már megírtunk.)

Szükség is van a nagyobb aktivitásra. Például a Corel másik jelentős „beolvasztott” terméke, a WordPerfect nálunk nem rendelkezik olyan erős hagyományokkal, mint az Egyesült Államokban, ezért a Corel WordPerfect Suite nem is nagyon terjed. A Ventura 8-asnak, mint a jelenlegi legsokoldalúbb integrált DTP szoftvercsomagnak a fent említett okokból Magyarországon jóval nagyobb esélye van. (Illetve inkább csak lenne, ha minden szempontból hozzáillesztenék a magyar nyelv követelményeihez.) Azt sajnos tudomásul

kell venni, hogy ezek a sokat tudó szoftverek erőforrást zabáló monstrumok. A Corel Ventura 8 futtatásához szoftverkövetelmény a Windows 95 vagy NT 4.0, a hardverben pedig bizony legyen CD-meghajtó, Pentium processzor, 32 MB memória és 65 MB szabad lemezterület.

Kedvező előjel viszont, hogy a Corel most már igyekszik kitörni a windowsos felület zártságából: megegyezett az Adobe céggel, hogy a Corel grafikai és DTP szoftvereibe integrálja a PDF szabványt; jelentős lépéseket tett, hogy beépítse programjaiba a HTML technológiát; a Sun partnereként vállalkozik Java-alapú alkalmazások készítésére; sőt elkezdte a Linux platformra történő fejlesztéseket is...

Az Areco virtuális irodaszer-áruháza

Ezen a nyáron — Magyarországon ilyen profillal elsőként — internetes irodaszer-áruházat nyitott az Areco. Hogy melyik Areco? Hát, az úgy van, hogy egy világszerte ismert kereskedelmi keretszoftverre, az iCat Electronic Commerce Suite-ra építve az Areco Systems Kft fejlesztette ki a konkrét alkalmazást az Areco Informatika Kft részére. Az iCat ideális eszköz azoknak, akik gyorsan és hatékonyan szeretnének kereskedelmi tevékenységet folytatni az Interneten. A program sok beépített funkciót és sémát tartalmaz, így azok a cégek, amelyek nagyon rövid idő alatt akarnak megjeleníteni a halózaton, akár néhány nap alatt megtehetik ezt. Az iCat főbb erői: könnyen kezelhető felület; beépített elemző-funkciók (a vásárlókra, a legkelendőbb termékekre, a rendelésekre stb.); saját nyelv az egyedi funkciók beépítésére; akár naponta más-más akciós ajánlat összeállítása; biztonságos fizetési módok használata.

A virtuális irodaszer-áruházban mindenki kedvére nézeledhet, anélkül, hogy regisztrálnia kellene magát, a vásárlás-

COREL VENTURA 8

Vector Editing
Edit imported vector images directly in Corel VENTURA 8, regardless of the original vector file format.

Graphics Export
Export pictures in one of over 30 supported graphic file formats.

Bitmap Effects
Manipulate images with over 70 bitmap Effects filters and complete Adobe® Photoshop®-compatible filter plug-in support; apply artistic effects without having to switch to another application.

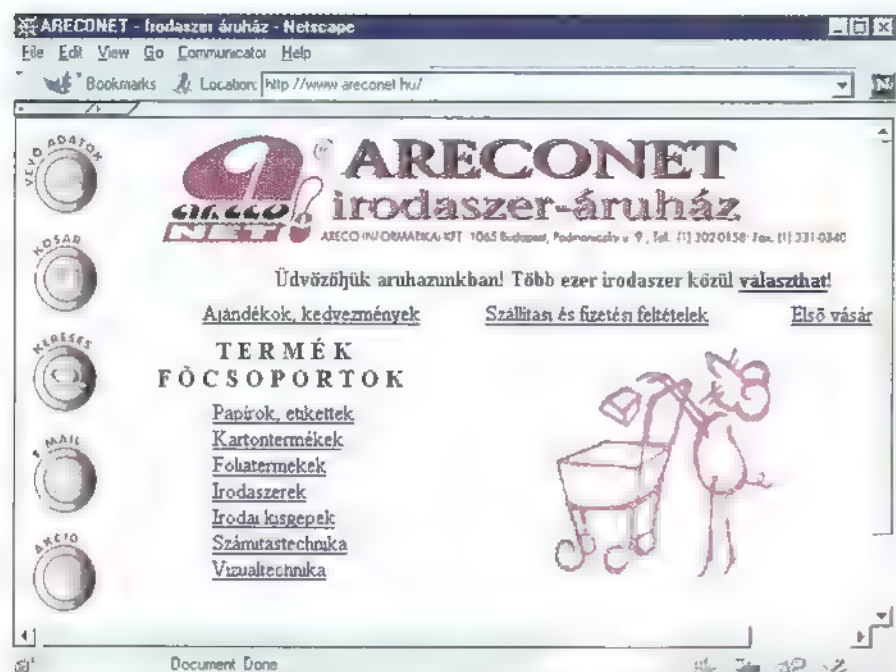
User-Defined Display Resolution
Drastically improve file display and load times by defining the display resolution of imported graphics.

Docker Windows
Enhance productivity by using drag-and-drop docker windows to maximize onscreen workspace.

Timely Power

VENTURA Navigator 1
Publication Manager

Name	Style	Type
Body Text	Not	Body Text
Header	Not	Header
Main Heading	Not	Heading
Minor Heading	Not	Heading
Numbered List	Not	List Item
Page #	Not	Page
Subheading	Not	Heading
Z-FOOTER	Not	Page
Z-HEADER	Not	Page
Z-SEC1	Not	Page
Z-SEC2	Not	Page



hoz természetesen már szükséges a szállítási és a számlázási cím megadása, valamint a vevői adatlap kitöltése. Vajda Péter, az Areco Systems ügyvezetője szerint a fejlesztéshez azért választották az iCat szoftvert, mert az rugalmassága révén egyedi funkciókkal is felruházható. Lehetőség van további biztonságos fizetési módok beépítésére is, ezért az Areco Systems azt tervezi, hogy amikor elhárulnak az akadályok az új SET (Secure Electronic Transaction) biztonsági szabvány alkalmazása előtt, az általuk készített online kereskedelmi rendszerekbe integrálni fogják és választható opcióvá teszik a hitelkártyás fizetési módot. (További információk a www.areconet.hu vagy a www.irodaszer.com címen.)

ManageWise 2.6

A Novell 1998. június 29-én bejelentette a ManageWise új verziójának béta-változatát. A rendszerfelügyeleti munkák megkönnyítésére a szoftver számos újítást tartalmaz. Bár a NetWare 5 szerverplatform végleges változata még nem jelent meg, a ManageWise 2.6 teljesen kompatibilis lesz az új operációs rendszerrel. Támogatja a Windows NT konzolt, az NDS-felügyeletet és az integrációt a Novell Z.E.N.works munkaállomás-felügyeleti megoldásaival.

A 2000. évi (és az utána következő) dátumok helyes kezelésére is felkészítették a termék komponenseit: a konzol, a szerver és a munkaállomás moduljait. A felhasználói rendszerek későbbi bővíthetősége érdekében az említett (Netware 5.0, Windows NT) konzolokon kívül támogatja valamennyi NetWare verziót és a Windows 95 programot. A felhasználó tehát szabadon összeválogathatja az adott ponton legjobban megfelelő operációs rendszerrel felszerelt eszközt, mert az új program lehetővé teszi azok egységes felügyeletét. (A béta-változat letölthető a <http://www.support.novell.com/beta/public> címről.)



NÉZD MEG... VÁLASZD KI... SZAVAZZ RÁ!

A figyelem most a hálóra irányul. A pályázatok beérkeztek, a zsűri értékeli, de amíg a döntés megszületik a közönségszavazás az idő. A szigorú szakmai szemle mellett, melynek nyertesei egymillió Ft-ot nyernek, várjuk az érdeklődők, ismerősök, szimpatizánsok, barátok, és ellenfelek, valamint a véletlenül erre böngészők szavazatait is.



<http://suli.gyaloglo.hu>

A közönségszavazás fődíját,
egy NOKIA mobil kommunikátort

PANNON GSM
Az élvonal.
ajánlotta fel.

Szavazz a legjobbra!

A pályázat kiírói:
CIPZŐR JÁNOS
APUKOM
világa

COMPUTER

GUARDS Web

A támogatók:

ORACLE®

Sun

Guards Távközlési Rt.
Internet Üzletág
1077 Budapest, Rottenbiller u. 33.
Telefon: 06-80 24 20 33
<http://www.guards.hu>

Webpályázat és ingyenes e-mail

Akik már hallottak róla, bizonyára „elgyalogoltak” a <http://egon.gyaloglo.hu> címre, hogy megismerkedjenek azzal az ingyenes, web-alapú elektronikus levelező szolgáltatással, amelyet a Guards Távközlési Rt. nyújt. Bárki saját e-mail címet kaphat, elektronikus leveleket fogadhat és küldhet, egyéni címlistát alakíthat ki, feliratkozhat levelezőlistákra vagy tallózhat a magyar levelezőlistákban. Hogyan térül meg egy ilyen befektetés egy cégnek? Valószínűleg a hosszú távú üzletpolitikában, tekintve, hogy a Guards sok egyéb területen (rádiótelefon, műholdas szolgáltatások, Internet, házimozi stb.) kamatoztatni tudja az ingyenes e-mail révén is erősödő szakmai presztízsét. Hasonlóképpen a jövő építését szolgálja a középiskolások részére meghirdetett weblapkészítő pályázat, melynek egyik kiírója és szervezője ugyancsak a Guards. A legjobbak kiválasztására a felkért tagokból álló „hivatalos” zsűri mellett közönségszavazás is lesz. Bővebb információ a fenti képen is látható <http://suli.gyaloglo.hu> webcímen.

Albacomp Rt.
8000 Székesfehérvár
Mártírok útja 9.
Tel.: (22) *315-414
Fax: (22) 327-532

Budapesti Kirendeltség
1065 Budapest
Frangepán u. 8-10.
Tel.: 12-91-493
Tel./fax: 12-90-152

Szaküzletek
1065 Budapest
Nagymező utca 25.
Tel.: 11-18-095
Tel./fax: 13-18-108

1011 Budapest
Tel.: 11-263-311
Tel.: 11-207-4404
Fax: 207-1392

3525 Miskolc
Rákóczi u. 19.
Tel.: (46) 354-263
Tel./fax: (46) 357-136

Az Albacomp Activa számítógépcsalád különböző Intel processzorokat tartalmaz



ALBACOMP
activa



PENTIUM® II
PROCESSZORRAL

AZ INTEL INSIDE EMBLEMA ES A PENTIUM AZ INTEL CORPORATION BEJEGYZETT VEDJEGYE. AZ MMX AZ INTEL CORPORATION VEDJEGYE

*N*agy figyelmet igényel a főszereplő kiválasztása. Minden pillanatban tökéletesnek kell lennie – rábíztuk, ami a legfontosabb. Csak pontos, magas színvonalon dolgozó, átlagon felül teljesítő lehet.

Mindent elronthatnak a gyenge statiszták is. Gondolataink hiába szárnyalnak, bukásra vagyunk ítélve, ha nem hibátlanul tesszük a dolgukat.

Ki ne értené, hogy milyen lényeges mindez...

ha számítógé



gépéről van szó

Digitális fényképezőgépek III.

A Kodak hobby-kínálata

A Kodak cég neve elsősorban a hagyományos, ezüstalapú fotóanyagok gyártásával és fejlesztésével kapcsolatban ismert. Időközben azonban a digitális fotótechnika területén is úttörővé vált. Készítenek egyszerűbb digitális „hobby” gépeket, és professzionális kategóriájú „still video” kamerákat is. Most az előbbi csoportból mutatunk be három jellegzetes típust. (A kamerákat kipróbálásra a Mikropo bocsátotta rendelkezésünkre.)

DC120 Zoom

Ez a gép „távcső” rendszerű konstrukció. Úgy kell megfogni és a szemhez emelni, mint egy távcsövet. A CCD felbontása 850x984 pixel, 24 bites színmélységgel. Háromszoros zoom objektívvel rendelkezik. A gyújtótávolság változtatásának tartománya megfelel a kisfilmes fényképezőgépek 38 és 114 mm közötti értékének. Előtétlencsékkel a gyújtótávolság tovább módosítható. A tele előtéttel megduplázható a gyújtótávolság, a nagylátószögű előtét 0,65-szeresére, a szuper nagylátószögű előtétlencse 0,5-szeresére változtatja a gyújtótávolságot. A közelfényképező előtétlencsével 9 cm-re is megközelíthetjük a témát. Az előtétek az optikai kereső látószögét nem befolyásolják, ezért ilyenkor csak a TFT LCD (vékony tranzisztortrétegű folyadékkristályos) kijelzőn ellenőrizhető a képkivágás.

Az automatikus élességállításnak két üzemmódja van, egy- és többobjektumos. Egyobjektumos módban a kép közepén lévő témát állítja élesre. Többobjektumos módban a gép megvizsgálja a képmező három pontján a téma távolságát, és az élességet ezek átlagolásával kapott közepes értékre állítja be.

A 20 és 50 cm közötti témáknál az élességet viszont kézzel kell beállítani. Alaphelyzetben a kép világosságát, az expozíciót is automatikusan szabályozza a gép. Lehetőség van azonban az expozíció korrigálására vagy kézi beállítására. A kézi beállításnál a rekesznyílást és a megvilágítási időt együtt változtatja, ami a túl világos vagy túl sötét háttérű témáknál, valamint külső vaku használatánál előnyös. A készülékbe 2 MB-os memória van fixen beépítve. Ezt egy behelyezhető Kodak Picture Kard memóriakártyával lehet bővíteni. Az alapmemórián a tömörítési aránytól függően 2, 7, 12 vagy 20 kép fér el.

Sötét helyen nemcsak a beépített kisvakuval fényképezhetünk, mert külső vakuhoz való szinkroncsatlakozót is találunk a gépen. Ez lehetőséget ad több villanókészülék, esetleg professzionális műtermi vakuberendezés használatára.

A készülék hátulján az optikai kereső mellett egy 41 mm átlóméretű színes LCD kijelző szolgál monitorként. Ezenkívül egy másik kis LCD-n (státuszjelzőn) láthatók a működés aktuális állapotát mutató információk. A színes LCD monitoron előnézeti üzemmódban már a rögzítés előtt megtekinthető a kép. Figyelembe kell azonban venni,

hogy az előnézet üzemmód jelentősen rövidíti az elemek élettartamát.

A fényképezés után megjeleníthetők a tárolt felvételek, egyenként is, de úgy is, hogy egyszerre 4, illetve 9 kép látszik. A memóriában tárolt képek digitális albumokba rendezhetők. A belső memória 8 ilyen albumot tesz lehetővé, a memóriakártya 60-at. Az albumok nevet kaphatnak, így a képek utólagos visszakeresése sokkal egyszerűbb. A képek KDC formátumban kerülnek a memóriába. Lehet tárolni tömörítés nélkül vagy háromféle tömörítési aránnyal. Az adatok soros porton keresztül vihetők be PC vagy Macintosh rendszerű számítógépekbe.

DC210 Zoom

Ez a digitális fényképezőgép külsőleg már nem távcsőhöz, hanem a hagyományos kompakt kamerákhoz hasonlít. Az optikai kereső mellett, a gép hátulján lévő színes monitor átlómérete 46 mm. A CCD felbontása 1152 x 864 pixel, 24 bites színmélységgel. Az objektív (kisfilmnek megfelelő) gyújtótávolsága 29 és 58 mm között változtatható. A 29 mm a gyakorlatban is jól használható nagylátószögű objektívet jelent. Az 58 mm viszont még nem tekinthető teleobjektívnek, a gyakorló fotográfusok tudják, hogy a teleobjektívet igénylő leggyakoribb témákhoz (portré, alakfotó) minimálisan 80-100 mm-es gyújtótávolság szükséges. Ehhez a típushoz egyenlőre sajnos nincsenek előtétlencsék. Makro üzemmódban az objektív 9 cm-ről is éles képet ad. A kép világosságát, az expozíciót lehet



korrigálni, de az automatika nem kapcsolható ki teljesen.

A DC210-be nem építettek belső memóriát. Egy 4 MB-os kivehető Kodak Digital Science memóriakártyával szállítják. Ezt ki lehet cserélni 8, 10 vagy 15 MB-osra is. A 4 MB-os kártyára a tömörítés mértékétől függően 13-60 kép rögzíthető. A képet nemcsak a CCD teljes pixelszámának megfelelően lehet elkészíteni, hanem be lehet állítani a felbontást 640 x 480 pixelre is. Ilyenkor is az egész CCD felület működik. Ebből szoftveres úton jön létre a VGA szabványnak megfelelő felbontású kép.

A képeket FPX vagy JPG formátumban lehet elmenteni különböző tömörítési arányokkal. A képpel együtt a gép különböző információkat is rögzít (dátum, időpont, beállítások). Ezeket a program a számítógépben a kép mellett külön ablakban képes megjeleníteni.

A Kodak DC210-nek videojel kimenete is van (PAL, NTSC), így a tárolt képek televízión is megnézhetők. A kép a számítógépbe a soros csatlakozáson kívül infravörös porton keresztül, vezeték nélkül is átvihető. A színes LCD képernyőn kellemes grafikus felhasználói felület segíti a legfontosabb funkciók vezérlését.

DC260 Zoom

A Kodak DC260 kamera saját kategóriájában kiemelkedő a CCD felbon-



tásával és szolgáltatásaival. A fizikai felbontás 1536 x 1024, azaz 1,6 millió pixel. Ez megfelelő nyomtatóval 20 x 25 cm méretű fotó minőségű nyomatot tesz lehetővé. Az objektív itt is háromszoros optikai zoommal rendelkezik. Van a gépen átnézeti optikai kereső, de emellett egy 2 coll átmérőjű TFT színes képernyő is helyet kapott a hátfalon. Találunk a kamerán szinkronzsinór csatlakozót külső vakuhoz. Ennek kü-

lönösen azok örülhetnek, akik figyelmet fordítanak az igényes világításra. Így nemcsak a gépen lévő kis villanókészüléket lehet használni, hanem például professzionális fotóműtermi vakustúdiót is. A külső villanófény igénybevételéhez az is szükséges, hogy a rekesznyílást az automatika kikapcsolásával manuálisan lehessen szabályozni. Itt erre is lehetőség van. A DC260 kamerának fixen beépített belső memó-

Technikai adatok

	DC120 Zoom	DC210 Zoom	DC260 Zoom
Fizikai felbontás	850 x 954 / 24 bit	1152 x 864 / 24 bit	1548 x 1032 / 24 bit
Interpolált felbontás	1280 x 960	640 x 480	1152 x 768 vagy 768 x 512
Objektív	38–114 mm	29 — 58 mm	38 — 115 mm
Élességállítási tartomány	50 cm-től végtelen	50 cm-től végtelen	33 cm-től végtelen
Makro üzemmód	20–50 cm	9 — 50 cm	-
Expozícióvezérlés	Automata (korrigálható) vagy manuális	Automata (korrigálható)	Automata vagy kézi beállítású
Zársebesség	1/16 — 1/500, manuálisan is állítható	1/2 — 1/362	1/4 — 1/400
Rekesznyílások	-	-	Nagylátószögnél 3 — 14, tele állásban 4,7 — 22
Beépített memória	2 MB	Nincs	Nincs
Memóriakártya	4, 8, 10, 15 MB	4, 8, 10, 15 MB	8, 10, 15, 32 MB
Kereső	Optikai és LCD képernyő	Optikai és LCD képernyő	Optikai és LCD képernyő
Színes LCD monitor	41 mm színes LCD, képnézési és szerkesztési funkciók	46 mm színes LCD, képnézés, szerkesztés, grafikus felhasználói felület	2 coll színes LCD, képnézés, szerkesztés, grafikus felhasználói felület
Vaku	Automata	Automata	Automata, lehetőség külső vaku csatlakoztatására.
Egyéb lehetőségek	Előtétlencsék, szűrők	Video kimenet (PAL, NTSC), két képformátum (JPG, FPX)	Video kimenet (PAL, NTSC), két képformátum (JPG, FPX)
Csatlakozó felület	Soros PC / MAC	PC / MAC soros, infravörös	PC / MAC soros, infravörös

riája nincs, de 8 MB-os memóriakártyával szállítják. Ez a jelenlegi kínálat szerint 32 MB-osra is kicserélhető.

A gépbe egy sokoldalú szoftver került. Érdekes lehetőség, hogy a funkciók átprogramozhatók. Így az egyes szolgáltatások egyéni igényeknek megfelelően alakíthatók ki. Ez az átalakítás egy saját karakteralapú nyelv segítségével történik.

Lehetőség van különböző automatikus képsorozatok készítésére. Ez a gyakorlatban azt jelenti, hogy a kioldógomb egy lenyomására gyors egymásutánban több felvétel készülhet. Előre be lehet állítani, hogy ezek miben különbözzenek egymástól. Például lehetnek különböző világosságúak (expozíciós sorozat) vagy különböző színűek (színegyensúly sorozat). A felvételek maximálisan 0,3 másodpercenként követhetik egymást. Az így készíthető képek száma a felbontástól függően 2 és 12 között lehet. Intervallum üzemmódban előre beállított időközönként automatikusan felvételeket készít. A képekre idő és dátum, valamint szöveges információ vihető rá. Az álló formátumban készült képeket a TFT képernyőn

Szoftvertámogatás

A Kodak DC kamerához négy program tartozik:

1) Kodak Picture Transfer Application

Lehetővé teszi a képek áttöltését a számítógépbe, és tárolását a legismertebb formátumokban (KDC, BMP, TIFF, JPEG, PCX, PICT).

2) Kodak Mounter

Képtámvivő és a gép beállításait módosító program. Lehetővé teszi, hogy a digitális kamera memóriáját a számítógép operációs rendszere úgy kezelje, mint egy meghajtót.

3) Kodak Digital Access

Segítségével a TWAIN-t támogató grafikai programokba behívhatók a képek a fényképezőgép memóriájából. A fényképezőgép beállításainak módosítására is lehetőséget nyújt.

4) PictureWorks PhotoEnhancer

Fotóretusáló, képmódosító program.

A DC210 és a DC260 fényképezőgépekhez ezenkívül mellékelik az Adobe PageMill (HTML szerkesztő) és az Adobe PhotoDeluxe (képmódosító) programokat is.

való megtekintésnél „talpra” állítja (Orientation Sensor). Ezért nem kell a gépet forgatni a visszajátszásnál. A memóriában lévő képek névvel ellátott „albumokba” rendezhetők. Így könnyebbé válik a visszakeresés.

A készülék alkalmas hangrögzítésre is, és ennek révén a fotókat hangos kommentárral láthatjuk el, a hangfelvételeket pedig a kamerával is visszajátszhatjuk.

Dékán István

Microsoft
**Technical
Education
Center**

... hogy a **legjobb**
helyeken dolgozhasson

Microsoft
Professional

Microsoft
Professional

Microsoft
Professional

Microsoft
Professional

Sales
Specialist

Microsoft

A weblap címe: www.hpo.hu

Szabadalmi „mindentudó”

A Magyar Szabadalmi Hivatalhoz érkező sokféle érdeklődést nagyon jól illusztrálja az alábbi levéldézet:

„Sajnos jogvédelemi és szabadalmi ügyekben nagyon tájékozatlan vagyok. Tapasztalataim azt mutatják, hogy manapság mégis érdemes foglalkozni ezzel a kérdéssel. Vállalkozásomban eljutottam oda, hogy tudnom kellene, vajon a termék, amelyet készítek, illetve amelynek a sorozatgyártását terveztem, jogvédelem alatt áll-e már valahol. Mivel családi vállalkozáson belül bútorgyártásról van szó, nem tudom, lehet-e az egyes darabokat vagy magát a termékcsaládot, a stílust védeni. Korábban előállított termékeink közül jó néhányat láttam már az ország különböző pontjain, de azok sajnos más gyártók műhelyéből kerültek ki. Kérdéseim is ezzel kapcsolatosak. Ha a termék még nem védett, és én szeretném azt oltalom alá helyezni, mi ennek a menete? Mekkora a várható költségek? Érdemes-e egyáltalán megtenni, azaz nyújt-e gyakorlati védelmet a mások által történő másolás, gyártás ellen? Hol tudom megkeresni a már meglévő védett termékeket, honnan tudom meg, védett-e már egy termék vagy sem?”

Annak, aki érdeklődik a bevezetőben idézett kérdések iránt, érdemes továbbolvasnia ezt cikket. A főbb kérdésekre iparjogvédelmi tanácsadóként adott válaszok mellett sajátos metszetét szeretném adni a Magyar Szabadalmi Hivatal weblapjának, illetve bemutatni annak használatát.

Vegyük tehát sorra a főbb kérdéseket! Hogyan lehet védeni például egy bútordarab vagy bútorcsalád stílusát, külső megjelenését? Ha ellátogatunk a <http://www.hpo.hu> címre, ott egymás mellett a Magyar Szabadalmi Hivatalban megszerezhető ötféle jogi oltalom nevét találjuk. Ezekre rákattintva hétköznapi nyelven, kérdés-felelet formában ismerkedhetünk meg az egyes oltalmakkal, és a megszerzésükre vonat-

kozó információkkal. Ezek közül a mi esetünkben az ipari mintaoltalom az, amely tárgyak formai kialakításának védelmét biztosíthatja.

Az első kérdésre, hogy hogyan lehet megszerezni az ipari mintaoltalmat, részletes és világos választ kapunk, ha az oldalon lefelé haladva rákattintunk a rajzocskára. Színes folyamatábra képe tárul elénk, amely a „Bejelentéssel” kezdődik. Erre a szóra kattintva az alsó ablakban megjelenik a hozzá tartozó magyarázat is. A hivatalhoz benyújtandó bejelentési kérelem formai és tartalmi követelményeit a folyamatábra jobb oldali sávjában található „Alaki előírások”-ra kattintva nézhetjük meg. A várható költségekre vonatkozóan a „Díjak” szóra kattintva találjuk meg a választ.

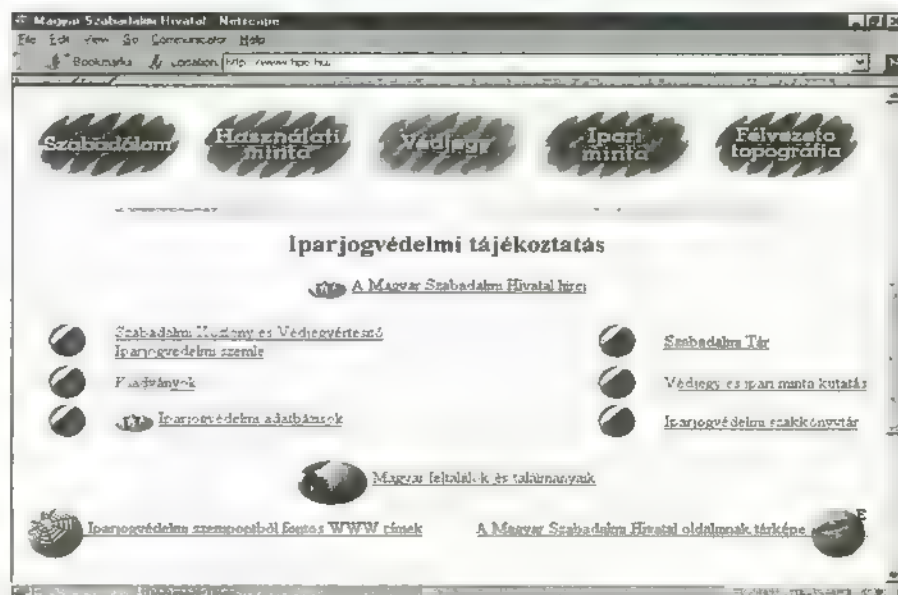
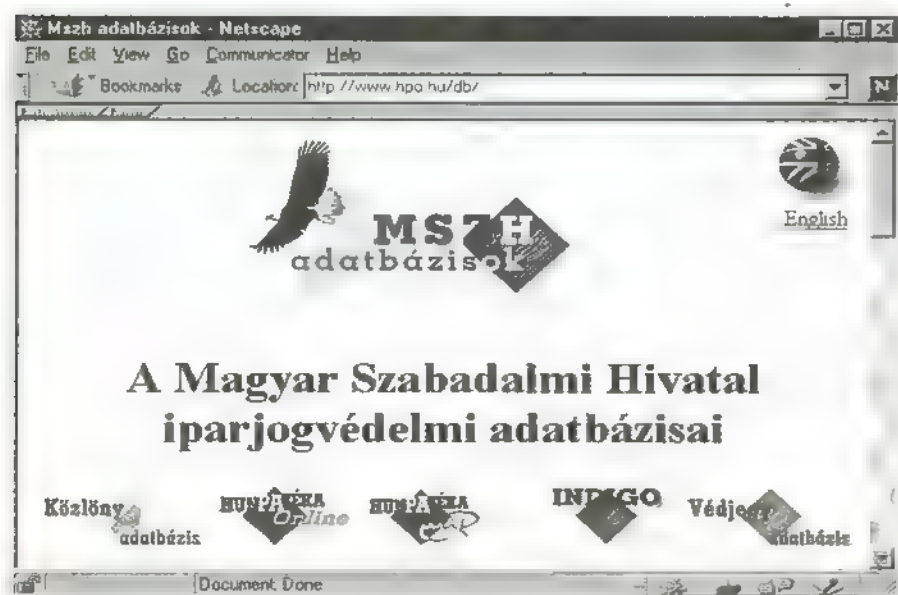
Arra is lehetőség van, hogy a „Hozzáad” gomb segítségével összegezzük a felmerülő költségeket.

Nyújt-e gyakorlati védelmet az ipari mintaoltalom a mások által történő másolás, gyártás ellen?

Ha az ipari mintaoltalmi oldal folytatását is megnézzük, ott az utolsó előtti kérdés foglalkozik azzal, hogy mi a teendő jogsértés esetén. Világosan kiderül, hogy az ipari mintaoltalom megsértése miatt a jogosult pert indíthat a bíróság előtt, követelheti a jogsértés megállapítását, a jogsértés abbahagyását, a jogsértést megelőző állapot helyreállítását, valamint a kártérítést.

Hol lehet keresgélni a már meglévő védett termékek között, honnan tudható meg, hogy védett-e már a termék vagy sem?

Ez alapvető kérdés. Mielőtt rászánánk magunkat, hogy elkészítjük a bejelentési iratot, és benyújtásakor befizetjük a bejelentési díjat, van egy fontos dolgunk: lehetőség szerint meg kell győződnünk róla, hogy legalább Magyarországon nem védi-e korábban bejelentett ipari mintaoltalom a kérdéses terméket — ezáltal felesleges kiadásoktól kímélhetjük meg magunkat. Az MSZH főlapján az „Iparjogvédelmi adatbázisok” címszónál válasszuk ki az „Indigo” ipariminta-adatbázist. A nyitóoldalon olvashatjuk, hogy ez az adatbázis csak az érvényes ipari mintaoltalmi dokumentumok és a folyamatban lévő ipariminta-bejelentések adatait és illusztráló ábráit tartalmazza. Ha a valaha érvényben volt összes ipari mintára vagyunk kíváncsiak, akkor a Magyar



Szabadalmi Hivatal „Ipariminta-kutatóját” személyesen kell felkeresnünk. A kutató címe és az ügyfélszolgálat időpontja is egy kattintással elérhető.

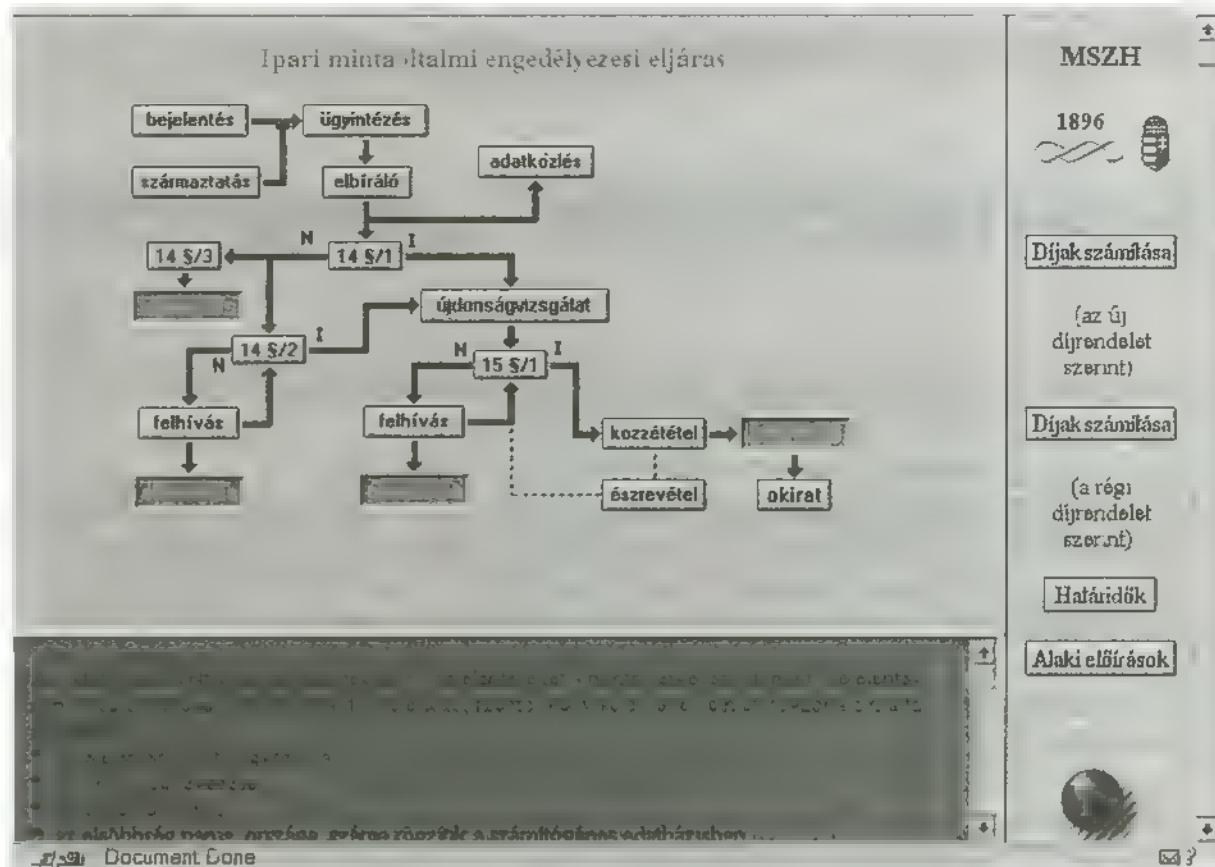
Nézzük meg, hogyan kell megkeresni kiinduló példánknak megfelelően a bútorokat az adatbázisban. A „Keresés az Indigo adatbázisban” kifejezésre kattintva egy űrlap tárul a szemünk elé. Kézenfekvő, hogy a „Cím szavai” mezőbe beírjuk a „bútor” szót, vagy a bútor szótövet bármely végződésre felkészítve hozzáillesztjük a * karaktert (bútor*). A keresőszó alapján azonban nem fogjuk megtalálni az egyes speciális bútorfajtákat, úgymint szék, asztal, járóka, gyerekágy stb., csak azokat, amelyekben a „bútor” és annak toldalékos alapjai szerepelnek. Hogy keresésünk teljesebb legyen, azaz lehetőleg minden bútorfajtaival kapcsolatos tételt megtaláljunk, a titokzatos nevű „Locarnói osztályozás” mezőt érdemes használni. Itt találhatjuk meg azokat az osztályozási jelzeteket, amelyek az ipari minták hordozóit, magukat a tárgyfeleségeket osztályozzák. Egy jelzet egy teljes fogalomcsaládot fed le. Kattintsunk a képernyő tetején vagy alján a „Locarnói osztályozás” nevű ikonra, illetve magára a mező nevére, és máris megjelennek az egyes tárgycsoportoknak megfelelő jelzetek. Az összes bútorfajta a 06. jelzet képviseli.

A kereső űrlapot kitöltve esetünkben 454 találatot sikerült elérni, míg a bútor szóval keresve csak 40 találatunk volt. A listából a számunkra érdekes tételeket az azonosító számok (bejelentési ügyszám vagy lajstromszám, attól függően, hogy megkapta-e már a minta az oltalmat vagy sem) előtti négyzetekre kattintva jelölhetjük ki, és a ... ikon segítségével nézhetjük meg teljes terjedelmében.

Ha a kutatást eredményesnek ítéljük, vagyis általunk védeni kívánt terméket nem találtunk az adatbázisban, akkor töltsük le vagy nyomtassuk ki a „Hogyan szerezhető ipari mintaoltalom?” kérdésre adott válaszban felkínált, a számunkra legmegfelelőbb formanyomtatványt. Utána a Magyar Szabadalmi Hivatalban megtehetjük az ipari minta bejelentését.

A weblapon a többi oltalmi forma — a szabadalom, a használati minta és a védjegy — esetében is hasonló információkra, folyamatábrákra, adatbázisokra számíthatunk. Sőt: sok egyéb hasznos, iparjogvédelemmel kapcsolatos információra is bukkanhatunk, például az „Iparjogvédelmi szempontból fontos WWW-címekre”.

Vadász Ágnes



MAGYAR IPARI MINTA ADATBÁZIS

Vélemény

Locarnói osztályozás



MSZH honlap

Sugó

adatbázisok

Legfrissebb adatok: 1997. november 30.

Cím szavai:	<input type="text" value="bútor"/>	Pl.: Illatszer* or kölni
Jogosult és szerző nevei:	<input type="text"/>	Pl.: Hasbro
Jogosult országa:	<input type="text"/>	Pl.: GB
Locarnói osztályozás:	<input type="text"/>	Pl.: 12-11
Bejelentés, közzététel, lajstromozás napja:	<input type="text"/>	19970101--
Bejelentési ügyszám:	<input type="text"/>	Pl.: D9600158 84832
Találatok száma:	<input type="text" value="Húsz"/>	Ékezetűtípus:
		<input type="text" value="Windows ékezetek"/>
		Publikációs fázis:
		<input type="text" value="Minden dokumentum"/>

Keres

VÉDJEGY ADATBÁZIS - Netscape

File Edit View Go Communicator Help

Bookmarks Location: http://www.hpo.hu/db/vedjegy/vedjegy.cg

Vélemény Nizzai osztályozás MSZH honlap Sugó

adatbázisok

Legfrissebb adatok: 1998. május 31.

Védjegy:	<input type="text"/>	Pl.: Ipari minta
Bejelentő, jogosult neve:	<input type="text"/>	Pl.: Acro-Net
Bejelentési ügyszám:	<input type="text"/>	Pl.: M9301620 141952
Bejelentés és lajstromozás napja:	<input type="text"/>	Pl.: 19970101--
Érvényes jogosult országa:	<input type="text"/>	Pl.: US
Nizzai jelzetek és árjegyzék:	<input type="text"/>	Pl.: ...
Találatok száma:	<input type="text" value="Tíz"/>	Ekezetűtípus:
		<input type="text" value="Windows ékezetek"/>
		Védjegy típusa:
		<input checked="" type="checkbox"/> ...
		<input checked="" type="checkbox"/> ...
		<input checked="" type="checkbox"/> ...
		<input checked="" type="checkbox"/> ...
		<input checked="" type="checkbox"/> ...
		Védjegy származása:
		<input checked="" type="checkbox"/> Magyar védjegy
		<input checked="" type="checkbox"/> ...

Document Done

Sun: Internet szoftverek

A Sun Microsystems cég két olyan terméket jelentett be, amelyek az Internet-szolgáltatók és a vállalati hozzáférés-szolgáltatók számára megkönnyítik a következő generációs levelezési és üzenetküldési szolgáltatások megbízható bevezetését és működtetését. A Sun Internet Mail Server (IMS) következő, és a Sun Internet Calendar Server (ICS) első verziója a skálázhatóság és a könnyű használhatóság új szintjét jelenti. Az IMS új, gyors készenléti opciója heti 7x24 órás hibátlan működést tud fenntartani. Rendszerbiba vagy az útválasztó karbantartása esetén a rendszer azonnal átkapcsol egy pótszerverre, és a hálózat zavartalanul tovább működhet. Ezáltal csökken a más szolgáltatókhoz átvándorló előfizetők száma, illetve több lesz az elégedett vállalati felhasználó. Mindkét termékben megtalálható a Sun WebAccess szoftver is. Ez az alkalmazás lehetővé teszi, hogy az e-mail és a naptárszolgáltatások HTML-en keresztül is elérhetők legyenek, így a felhasználók bárhol is elolvashatják leveleiket.

A két alkalmazás további szolgáltatásai: könnyű adminisztráció (akár kétféle postaláda szervertől, a felhasználók és a szerverek felkészítése a legnagyobb Internet-szolgáltatók és vállalatok igényeire is); könnyű integrálhatóság (használható minden elterjedt e-mail klienssel, így a Microsoft Exchange-dzsel is, kapcsolódás olyan levelezési rendszerekhez, mint a PROFS, a cc:Mail vagy az MS Mail); biztonság (üzenettitkosítás a kliens és a szerver között, az Entrust Certificate Authority Server támogatása); „anti-spam” képességek, naptározás és előjegyzés (a személyes és a csoportnaptárak kifinomult előjegyzési rendszert biztosítanak).

Cisco: IP + ATM együtt

Olyan új hálózati termékeket jelentett be a Cisco Systems, amelyek az IP alapú szolgáltatások új generációját jelentik az IP és az ATM technológiák tökéletes integrálásával. A szolgáltatók IP + ATM alapú szolgáltatásokat vezethetnek be ezekkel az új eszközökkel, amelyek mindegyike tartalmazza a Cisco IOS operációs rendszerét és szolgáltatáskezelő eszközeit. Az új WAN hálózatperemi eszközök a most megjelent MPLS (Multiprotocol Label Switching) szabvány első megvalósításaként Tag Switching révén integrálják az IP és ATM technológiát. A három eszköz közül két új kapcsoló (BPX 8650 és MGX 8800), valamint egy általános szolgáltatási csomópont (BPX 8680) kerül a piacra. A BPX 8650 tartalmazza az ATM hálózaton IP kapcsolást lehetővé tevő MPLS-t. Az MGX 8800-at a kis és nagy ellátási területek és a vállalatközpontok igényeinek kielégítésére fejlesztették ki olyan helyekre, ahol a szolgáltatóknak akár 1400 DSI csatlóóra van szükségük. A BPX 8680-at pedig az 16 ezer csatlót is igénylő nagyon nagy méretű helyszínek számára fejlesztette ki a Cisco. Az átfogó IP + ATM megoldáskínálat részeként a Cisco piacra dobta a telekommunikációs infrastruktúra felépítéséhez pozicionált TGX 8750 jelű optikai hálózatmag-kapcsolót. Ez teljes mértékben MPLS-kompatibilis, és funkciói közé tartozik a hierarchikus PNNI (Private Network to Network Interface) protokollú csatló, a Sonet/SDH automatikus védett kapcsolás (Automatic Protection Switching) és a PC-48c optikai hálózat-összekapcsolás támogatása. Az új WAN hálózatmag-kapcsoló tovább csökkenti a szélessávú szolgáltatások nyújtásának költségeit, méretezhető, támogatja a különféle útválasztó protollokat. Olyan integrált szolgáltatásmenedzsment eszközt is fejleszt a Cisco, amely egyrészt a távközlési szolgáltatóknak nyújt további lehetőségeket, másrészt a más hálózati eszközszállítókkal való együttműködést is támogatja (például Newbridge).

HP: új stratégia, új disztribútor

Proactive Networking (PN) néven a hálózati rendszeradminisztrátorok összetett feladatainak segítésére, problémáinak megoldására olyan új koncepciót dolgozott ki a Hewlett-Packard,

amely a termék funkcionalitásának, az oktatásnak és a garanciális szolgáltatásnak a kombinációja. A PN-t a hálózatfelügyelet bonyolultsága és az üzemeltetési költségek csökkentése érdekében hívta életre a HP. A PN automatikusan detektál alapvető hálózati rendellenességeket, feltérképezi a hálózati topológia hibáit, felismeri a tipikus hálózatkábelezési hibákat, és ha a hiba a hálózat teljesítményét jelentősen befolyásolja, önállóan intézkedik. A HP PN a Network Performance Advisor segítségével feltárja a teljesítménynövelés lehetőségeit, és annak megoldási módjára javaslatot tesz. A HP Proactive Networking elemei: AdvancedStack hubok, ezen belül ValueLine Ethernet hubok (8, 16, illetve 12 és 24 portszámmal), FeatureLine rétegelhető (stackable) Ethernet hubok (12 és 24 porttal), AdvanceStack 100Base T 12TX és 24TX rétegelhető Fast Ethernet hubok (12, illetve 24 porttal), 100 VG AnyLAN hubok. A HP 200, 800, 2000 típusú AdvanceStack Switch munkacsoportos, illetve gerinchálózati kapcsolók. A ProCurve kapcsolócsalád a nagy hatékonyságú 1600M és 8000M típusú gerinchálózati ún. cut through Layer3 switchből áll. Az 1600M Gigabit Ethernet összeköttetéssel is felszerelhető, továbbá VLAN-ok felépítésére is alkalmas az IEEE 802.1Q szabvány szerint. A 8000M beépíthető redundáns tápegységgel is rendelkezik, VLAN-ok felépítésére szintén alkalmas.

3Com zászlóshajó: CoreBuilder 9000

A 3Com cég új CoreBuilder 9000 Enterprise Switch új generációs nagyteljesítményű hálózati kapcsolója egyaránt támogatja az ATM, a Gigabit Ethernet és az ASIC technológiát. A 3Com olyan alkalmazásokra is pozicionálja a berendezést, amelyek ATM alapú campus gerinchálózatot, a szerverekhez és a LAN munkacsoportok aggregálásához pedig Gigabit Ethernet és ATM alapú WAN hálózatot egyaránt feltételeznek. A berendezésbe épített Layer2 és Layer3 vezérlőfunkciók révén megszüntethetők a teljesítménycsökkentő szűk keresztmetszetek és a ma használt router alapú gerinchálózatoknál ismert topológiai hátrányok. A 3Com szerint az úgynevezett core switching területen a CoreBuilder 9000 nyújtja a legnagyobb kapcsolási teljesítményt. Skálázható architektúrája révén a berendezés max. 112 darab OC-12c (622 Mbit/sec) ATM portot, vagy 126 darab Gigabit Ethernet portot képes kiszolgálni. A Layer2 alapú adatcsomag-átvitel meghaladja a másodpercenkénti 100 millió csomagot (pps). Ugyanez az érték Layer3 kapcsolásnál elérheti az 56 millió pps-t. A CoreBuilder 9000 kiterjedt Quality-of-Service/Class-of-Service (QoS/CoS) képességekkel is rendelkezik. A 3Com a berendezést a nagy rendelkezésre állást követelő hálózati környezetekre pozicionálja. Adatcenterekben alkalmazva a nagyteljesítményű kapcsoló multitechnológiás, úgynevezett „collapsed” gerinchálózati eszközként alkalmazható.

Psion: Novell GroupWise a zsebben

Az angol Psion cég teljesen testre szabható komplett jogi megoldáscsomagot dobott piacra Series 5 kézi számítógépére alapozva. A megoldás lényege, hogy az ügyvédek a világ minden táján elérhessék a számukra fontos információkat, akkor is, ha mindenféle irodától távol vannak. A maximálisan 18 Mbájt memóriával rendelkező Psion Series 5 tartalmazza a Novell GroupWise-hoz való kapcsolódást elősegítő LinkWise és POP3 programokat, valamint az ügyvédi munkához szükséges, számlázási és egyéb funkciókat tartalmazó Timeslips programcsomag mobil gépes kiterjesztését, a Logit szoftvert. A Psion megoldásának fő célja, hogy a Series 5 révén az ügyvédek desktop gépeiben lévő információk hordozhatók, illetve távolról könnyen elérhetők legyenek. Az Advances cég LinkWise nevű, speciálisan a Psion Series 5-höz fejlesztett szoftvere a Novell GroupWise csoportmunkát támogató programcsomagját használóknak szinkronizáló eszközként szolgál. A Psion POP 3 e-mailes kliensprogramja segítségével pedig a GroupWise e-mail funkcióinak távoli elérése válik lehetővé.

Kovács Attila

Hálózati vetélkedő 5.: Kooperáció

Bővíthető rendszer



Napjainkban Magyarországon szinte minden vállalat számítástechnikai rendszere heterogén, és az informatikai rendszert a legkülönbözőbb gépek hatékony kooperációja teheti jól működővé. Így a hálózati operációs rendszerek választásánál nagyon fontos szempont az, hogy a kapott szerver gép jól illeszkedjen a meglévő struktúrába, ugyanakkor jó alapot képezzen a későbbi bővítésekhez. A NetWare hálózatok pontosan ilyen alapinfrastruktúrát adnak, és megteremtik a csatlakozási lehetőségeket különböző típusú munkaállomások és szerverek, más hálózatok (Internet), illetve a sokféle szoftvergyártó alkalmazásai felé.

A hálózatban az együttműködés alapvető feltétele a közös protokollok használata. A Novell NetWare támogatja a legfontosabb hálózati protokollokat. Az IPX/SPX támogatás magától értetődő, hiszen ez a protokoll éppen a piacvezető Novell által kifejlesztett szabvány a lokális hálózatokban. Az Internet terjedésével a TCP/IP és az ehhez kapcsolódó szolgáltatásokat megvalósító protokollok szerepe egyre nő. A NetWare is támogatja az összes, ezen a területen használatos szabványt (IP, TCP, UDP, DNS, NIS, FTP, HTTP stb.), tehát akár az Internethez szeretnénk csatlakozni, akár egy vállalati intranet kialakítása a cél, a NetWare mindenképpen ideális megoldás.

Munkaállomás-támogatás

Egy NetWare hálózat erőforrásaihoz a legkülönbözőbb munkahelyekről hozzáférhetünk. Létezik kliens program OS/2, Macintosh, DOS/Windows, Windows 95, Windows NT platformokon egyaránt. A magyar piacon különösen fontos, hogy akár egy kis teljesítményű boot-epromos, lokális diszk nélküli gép esetén is elérhetők a NetWare szerver szolgáltatásai. Emellett az is nagyon lényeges, hogy a kisebb teljesítményű és a régebbi operációs rendszert használó munkaállomások esetén is biztonságos legyen a hálózat. Egy intraNetWare-re épülő hálózat C2 biztonságú hálózati infrastruktúrát jelent, így teljesen mindegy, hogy a felhasználó a fent említett kliensek közül melyiket használja, az infrastruktúra biztonságos rendszerhasználatot teremt számára.

NetWare alapú hálózatunkba gond nélkül be tudunk illeszteni más szerver operációs rendszereket. A különböző

Unixok felé kialakítandó kapcsolat támogatására több modul is létezik.

Csatlakozás más szerverekhez

A fájllelés és a fájlmegeosztás az NFS protokollon keresztül biztosítható. A kétirányú nyomtatás a NetWare-Unix Print Services modulval valósítható meg. A rendszerek menedzsmentje, a felhasználók és jogosultságaik kezelése is „egy kalap alá tehető” az NDS (Novell Címtár — Novell Directory Services) révén. Gyakorlatilag minden Unix gyártó (SCO, HP, SUN, IBM) licencelte az NDS technológiát, így Unix verziók gond nélkül beilleszthetők egy NetWare hálózatba. A közös protokollhasználat mellett az NDS technológia a másik kulcs a különböző rendszerek együttműködéséhez. Az NDS nem más, mint egy központi adatbázis, amelyben tárolható az informatikai rendszer összes adata, a felhasználóktól a számítógépeken keresztül az alkalmazásokig. Egy NDS-alapú hálózat esetén a felhasználók egyetlen bejelentkezés után hozzáférhetnek a rendszer összes erőforrásához függetlenül attól, hogy azt melyik szervergép szolgáltatja, az adminisztrátorok pedig egyetlen program segítségével tudnak menedzselni akár egy heterogén, országos hálózatot is.

Az NDS révén Windows NT szerverek és munkaállomások integrációja is megoldott a Novell Workstation Manager, illetve az NDS for NT felhasználásával. A két modul alkalmazása szükségtelemmé teszi a nagyobb hálózatok esetén nehézkes Domain struktúra használatát. Az NDS-be integráltság révén a munkaállomások, a felhasználók, a jogosultságok, a policy-k menedzselése egy helyen történhet NetWare,

NT, Unix esetén. Egy NetWare hálózatból nemcsak NT és Unix alapú gépek felé nyílik kapcsolódási lehetőség. IBM Mainframe gépekhez és AS400 alapú szerverekhez is tudunk csatlakozni egy NetWare kliens gépről. A csatlakozás jelenthet egyrészt terminálemulációt az ottani alkalmazás futtatásához, másrészt nyomtatási lehetőséget ebből az alkalmazásból a NetWare hálózatban található nyomtatóra.

Alkalmazások integrálása

Az NDS, a Novell Címtár alapinfrastruktúra a vállalati hálózatok számára. A címtár kapcsolódási pontjai nyilvánosak, így bárki tud ehhez illeszkedő szoftvert fejleszteni. A Novell az Open Solutions Architecture (OSA) keretében foglalta össze az intelligens alkalmazások fejlesztéséhez nyújtott támogatást.

Az OSA révén a Novell biztosítja azt, hogy a fejlesztők könnyen tudjanak külön szolgáltatásokat hozzátenni termékeikhez. Az OSA adja a felhasználó azonosítását, a jogosultságok kezelését, az adminisztrációs programot, amely könnyedén kibővíthető az újonnan írt alkalmazás felügyeletéhez szükséges menüpontokkal. Már most is számos olyan alkalmazás létezik, amely kihasználja az OSA, vagyis lényegében a Novell Címtár által nyújtott lehetőségeket. Ilyen például a Novell GroupWise, a Z.E.N.works, a BorderManager. Természetesen nemcsak a Novell szoftverek építenek erre a struktúrára, hanem sok más gyártó terméke, például az Oracle adatbáziskezelője vagy a Cheyenne FAXserve szoftvere is kihasználja az NDS alapú hálózatok előnyeit.

Az együttműködés a Novell számára olyan hálózati infrastruktúra kiépítését jelenti, amely egyrészt biztosítja a számítógép hatékony használatát és felügyeletét, másrészt amelyben a rendszer a vállalat igényeinek megfelelően a legkülönbözőbb gyártóktól származó hardver és szoftver elemekkel könnyen bővíthető. Ez az infrastruktúra pedig jelenleg az NDS-sel a Novell címtár-szolgáltatásával alakítható ki a legkönnyebben.

Hargitai Zsolt

Hálózati vetélkedő 5.: Kooperáció

A csapatjátékos

A Microsoft Windows NT Server 4.0-t úgy tervezték, hogy valamennyi fontosabb ügyfél és hálózati kiszolgáló operációs rendszerrel együttműködjön, lehetővé téve az informatikai beruházások értékének megőrzését. A Windows NT Server további termékek megvásárlása nélkül támogatja a Unix, a Novell NetWare, az Apple Macintosh és a Microsoft Windows alapú hálózatokkal való együttműködéshez szükséges szabványokat. Elérhető Windows NT Workstation, Windows 9x, Windows 3.x, Mac OS, Unix, OS/2 és MS-DOS ügyfélgépekről, tartalmazza a fontosabb hálózati protokollokat (TCP/IP, IPX/SPX, NetBEUI, AppleTalk, DLC, HTTP, SNA, PPP, PPTP.)

A legfontosabb kérdés egy már meglévő hálózat és a Windows NT összekapcsolása. Az elsősorban hagyományos fájl- és nyomtatógéposztást nyújtó Novell NetWare környezetbe az NT simulékonyan illeszkedik be szolgáltatásaival. A meglévő NetWare kliensek változtatás nélkül használhatók Windows NT környezetben: a File and Print Services for NetWare (FPNW) szolgáltatás NetWare 3.x fájl- és nyomtatókiszolgálót szimulál. Ha windowsos hálózatban szeretnénk elérni egy NetWare szerver erőforrásait, akkor az NDS-t is támogató Gateway Services for NetWare beépített szolgáltatással a Windows NT Server átjárót kínál a NetWare Core Protocol (NCP) kliens oldali használata nélkül. A Directory Service Manager for NetWare oldja meg a vegyes hálózat egyszerű menedzselését. Hatékony támogatást tartalmaz az áttéréshez NetWare hálózatról a Migration Tool for NetWare szolgáltatás, amely a felhasználói fiókokat, a felhasználók adatait, állományait és jogosultságait is áthelyezi a Windows NT szerverre.

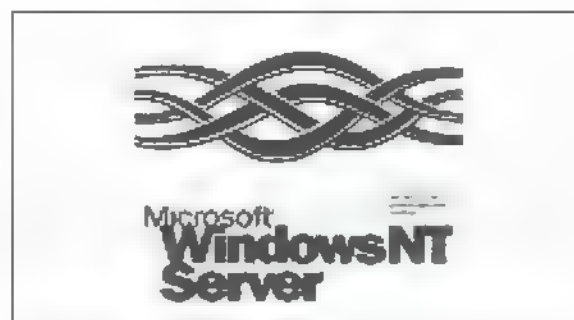
Mivel még ma is igen nagy számban használnak Unix alapú megoldásokat, elengedhetetlen, hogy költségkímélő és széles skálájú együttműködés alakuljon ki közöttük. Az együttműködés teljes, tehát nemcsak hálózati, hanem az adat-elérés, az alkalmazások futtatása és a menedzselhetőség is megoldott. A Microsoft a közelmúltban jelentette be a Services for Unix terméket, amely például Network File System (NFS) kliens/szerver szolgáltatást, telnetes távoli adminisztrációt, jelszó-szinkronizációt és kibővített parancsfájl-támogatást tartalmaz. Más cégek is kínálnak hatékony NFS kliens/szerver terméke-

ket Windows NT-re (Hummingbird NFS Maestro, NetManage ChameleonNFS/X, Intergraph DiskAccess és DiskShare).

A másik oldalról nézve a unixos munkaállomás a Samba démon telepítésével oszthatja meg erőforrásait a windowsos hálózattal. A Unix rendszerek által használt DNS DHCP, BootP és RPC funkciók a Windows NT Server beépített komponensei, a hálózati nyomtatáshoz pedig LPR klienst is tartalmaz. A kiszolgáló SNMP eszközzel is menedzselhető. Külső gyártók szoftverei is szép számmal fellelhetők ezen a területen (HP OpenView). Az X Window szerverszoftverek a Windows NT szervert valódi többfelhasználós hálózati operációs rendszerre alakíthatják (NetManage ChameleonNFS/X, Hummingbird Exceed).

Fordított eset, amikor egy Unix munkaállomáson szeretnénk Win32 alapú alkalmazást futtatni. Erre is szolgál a Microsoft egyik legújabb terméke, a Windows NT Server, Terminal Server Edition. (Ezzel a termékkel a régi, MS-DOS alapú számítógépeken is futtathatjuk a 32 bites Windows alkalmazásokat.) A Windows NT a kezdetektől fogva Symmetric Multiprocessing (SMP) megvalósítású, és támogatja a Posix szabványt, amely elősegíti a Unix alapú megoldások átültetését Windows NT-re (példa erre a Softway Systems OpenNT-je). A legnépszerűbb Unix alkalmazások (AWK, Perl, EMACS stb.) rendelkeznek Windows NT-n futó verzióval is, így az ezekre épülő (például Web-alapú) alkalmazások könnyen átültethetők Unixról Windows NT-re.

És még lehet fokozni: többplatformos alkalmazásfejlesztés? Igen, a Win-



dows NT beépített, komponensalapú alkalmazásfuttató környezetének alaptechnológiája, a Component Object Model (COM) már nemcsak a Windows platformokon elérhető, a COM for Solaris 1998 júniusától kész. Vezető rendszerintegrátorok (Andersen Consulting, EDS, KPMG, Vanstar) a COM segítségével egyesítik ügyfeleik többplatformos rendszereit. A Distributed Component Object Model (DCOM) pedig az elosztott megoldások létrehozásában segít. Adatbázis-elérés szintjén IBM DB2, Oracle, Sybase, Informix és más adatbázisokhoz elérése az ODBC és OLE DB szabványon alapul. A Windows NT Option Pack óta pedig operációs rendszer szintjén képes tranzakciókezelésre a Microsoft Transaction Server (MTS) segítségével. Az IBM CICS és IMS tranzakciók támogatása Transaction Serverrel, a Microsoft BackOffice Server csomagjába tartozó SNA Serveren keresztül történik.

Elosztott alkalmazások futtatása egy magánhálózaton vagy az Interneten keresztül, az Internet Information Server (IIS), az MTS és a Message Queue Server (biztonságos, várakozási sorokon alapú adattovábbítás-kiszolgáló) integrációjával valósul meg. Vegyes számítógépes környezetben az erőforrásokat a már több mint egy éve publikus Active Directory Services Interface-en keresztül egyszerűen kezelhetjük, legyen szó akár NDS, NT Directory Services (NTDS) vagy LDAP használatáról. Az ADSI-ra épülő Microsoft alkalmazások a Windows NT 5.0 megjelenése körül várhatóak, de például a Cisco hálózati eszközeit is e felületen menedzselhetjük majd.

A Windows NT lehetővé teszi, hogy a beépített titkosítási módszerek mellé saját elgondolásokat is megvalósítsunk a Security Support Provider Interface (SSPI) használatával. Ezzel nem zárjuk ki, hogy más rendszerek eljárásait integrálhassuk, hanem jelentősen növeljük az együttműködési képességet. Üzenetkezelés terén a Microsoft Exchange Server az LDAP protokoll mellett minden jelentős szabványt támogat: POP3, IMAP4, S/MIME, NNTP, X.400, X.500.

Csató Endre

Hálózati vetélkedő 5.: Kooperáció

Kényelmes átjárók



Mint előző cikkeinkben már említettük, az IBM-nek több kiszolgálóplatformja létezik. Legalsó, belépő szintű kiszolgálója az OS/2 Warp Server, amely könnyen tud kapcsolódni egyrészt a magasabb szintű, másrészt a már meglévő kiszolgáló platformokhoz.

Bizonyára nem árulok el nagy titkot, ha azt mondom, napjaink talán legelterjedtebb PC-alapú kiszolgálója a Novell NetWare. Természetesen az OS/2 Warp Servernek kötelessége ehhez a meglévő szerverhez kapcsolódni, illetve az átállást biztosítani. A Warp Server ehhez többféle segédeszközzel is rendelkezik.

Négy megoldás

Az első, talán legkézenfekvőbb megoldás a NetWare Gateway funkció, amelynek segítségével az OS/2 Warp Server egyetlen felhasználóként be tud lépni a NetWare szerverekbe, és az onnan kapott erőforrásokat (meghajtókat, könyvtárakat) tovább ajánlja saját klienseinek. Így a kliensek csak a Warp Serverre lépnek be, és ők csak a Warp Server erőforrásait látják, de ebben már benne lehetnek a Warp Server NetWare szerverektől kapott és tovább ajánlott erőforrásai.

A kliensek valójában nem is tudják, hogy az így kapott erőforrás egy NetWare szerverhez kapcsolódik. Sőt, a klienseknek nincs is szükségük NetWare támogatásra (IPX/SPX), a kliensek csak a Warp Server kliensprogramját használják.

Egy példa a könnyebb érthetőség kedvéért: a Warp Server belép mint felhasználó (például supervisor) a NetWare szerverre (vagy akár többre is), és jogosultságainak megfelelően „map”-eli magának F: meghajtóként a NetWare SYS:\ kötetét. Majd a Warp Serveren létrehozunk egy erőforrás-megosztást, ami a már „lokális” F: meghajtót osztja meg klienseinkkel. Ebből a példából láthatjuk, hogy a más hálózatokból kapott erőforrásokat a Warp Server képes továbbosztani kliensei számára. Ezért ezt gateway (átjáró) funkciónak is szoktuk nevezni.

Ha úgy döntünk, hogy régi NetWare szerverünket teljes egészében Warp Serverrel akarjuk felváltani, akkor erre

a célra a Warp Server csomagjában található egy segédprogram, és azzal könnyedén átvihetjük a NetWare-en lévő felhasználó-, csoport- és erőforrás-definíciókat Warp Serverre.

A harmadik megoldás az lehet, hogy bár a Warp Servernek külön DOS-os kliense van, az együtt tud működni a NetWare kliensekkel. Ez, mint tudjuk, más platformokon (OS/2, Windows-ok) nem jelent gondot, hiszen ott beépített kliensek vannak. Így a kliensek egyidejűleg használhatják a Warp Servert, és közvetlen kapcsolatuk nyílik a NetWare irányába is.

Bár nem a Warp Server érdeme, létezik egy negyedik megoldás is, és azt a Novell NetWare for OS/2 termék biztosítja. Ez gyakorlatilag maga a NetWare szerver, amely OS/2 operációs rendszeren fut. Így akár azt is megtehetjük, hogy ugyanazon a gépen futtatjuk az OS/2 Warp Servert és mellette magát a NetWare szervert.

Alias

Integrálhatjuk a Warp Servert a Windows platformokkal is. Itt sokkal könnyebb a dolgunk, hiszen a Windows platformok ugyanazt az SMB (Server Message Block) protokollt használják, melyest az OS/2 és a Warp Server. Itt tehát nincs szükség külön kliens vagy gateway programokra, hiszen a különböző operációs rendszerek mind a két irányban átjárhatók. Bár az IBM készített külön klienset Windows 95 és Windows NT platformokra, mivel a Warp Server képes erőforrásait ún. „alias”-okkal ellátni, így a felhasználó egyszerűen csak az aliasra hivatkozik, ha egy erőforrásra hivatkozni szeretne, és nem kell tudnia, hogy erőforrása melyik szerveren helyezkedik el.

Ennek előnye még, hogy a rendszer-adminisztrátor az erőforrásokat könnyen megváltoztathatja, áthelyezheti másik szerverre, anélkül, hogy az alias megváltoztatná. Így a kliensek, mivel

ők csak az aliasokra hivatkoznak, észre sem veszik a változást. A Windows platformok az aliasokra való hivatkozást azonban nem tudnák kezelni, az IBM készített a Windows platformokhoz külön kliens programot (mely teljesen ingyenes), és annak segítségével a Windowsok is képesek lesznek a Warp Server aliasait kezelni, együttműködve a meglévő funkciókkal. Tehát mind a Warp Server, mind a Windows NT teljes kooperációban tud együttműködni.

Direkt kapcsolat

Létezik az OS/2 Warp Serverhez NFS támogatás is, amellyel összekapcsolhatjuk a Warp Servert Unix rendszerekkel. Az NFS támogatás kétoldalú, tehát a Warp Server kliens is, szerver is lehet, így el tudja érni a Unixok fájlrendszerét, és meg is tudja osztani a sajátját a Unixokkal. Természetesen itt is érvényes, amit a NetWare Gateway funkciónál említettem, hogy a Warp Server egy Unixtól kapott erőforrást is tovább tud ajánlani saját kliensei számára, így a klienseknek nincs szükségük saját NFS támogatásra.

Bár az IBM Unix alapú AIX operációs rendszerére létezik direkt Warp Server támogatás is, amellyel nem NFS, hanem direkt kapcsolatot létesíthetünk a két kiszolgálóplatform között, a más Unixokon elterjedt Samba szintén használható a Warp Serverhez való kapcsolódáshoz. Ugyanez igaz az IBM AS/400 platformjára, ahol létezik a Warp Server for AS/400 termék, szintén direkt átjárhatóságot biztosítva az AS/400 és a PC platform között. Hasonlóképpen a LAN Server for MVS a mainframe-ekkel biztosít kapcsolatot.

Kliensek

Nem szóltam eddig a kliensekről. A Warp Server kliense lehet DOS, OS/2, Windows 3.1, Windows for Workgroups, Windows 95, Windows NT és Macintosh. Mint említettem, a Windows 95 és a Windows NT platformokon nincs szükség külön kliensre, mivel a Microsoft megoldása kompatibilis a Warp Serverével, de használható az IBM kiegészítése, amivel a Warp Server extra funkcióit is elérhetjük.

Pál Ferenc

Hálózati vetélkedő 5.: Kooperáció

Integráló erő a hálón



Az új Unixware7 lényeges tulajdonsága, hogy széles körben integrálható a hálózatokban kialakult különböző környezetekkel, platformokkal, harver- és szoftverrendszerekkel. Sorozatunknak ebben a részében ezeket az együttműködési lehetőségeket vesszük sorra.

Más Unix

Más Unix rendszerekhez való kapcsolódás természetesen nem jelenthet gondot, hiszen a jól bevált szabványok (TCP/IP, NFS, PPP stb.) alapján a Unix rendszerek jól megértik egymást, az integráció gyakorlatilag zökkenőmentesnek mondható.

Novell

A Unixware2 erős novelles alapjaira építve a Unixware7 tartalmazza a NetWare fájlkezelési, nyomtatási és könyvtári szolgáltatásait. Ezek a 4.10a verzióknak megfelelő szolgáltatásokat nyújtják, mind a kliens, mind a szerver oldaláról.

A legújabb Novell Directory szolgáltatásait (NDS 5.04) szintén támogatja a rendszer. A Netware szolgáltatások IP kapcsolaton keresztül is elérhetők a UnixWare IPX tunnelling segítségével.

NT

Ahol az NT szerverekkel való integráció lényeges szempont, ott a Unixware7-en futó Advanced File and Print Server csomag kiváló megoldást biztosít. Itt lehet fájl- és nyomtatómegosztást végezni Windows 95, Windows NT, Windows 3.x, OS/2 és DOS munkaállomásokkal.

A Unix könyvtárak és nyomtatók Windows hálózati meghajtóként és nyomtatóként látszanak. Természetesen domain-controllerként is tud működni. Mivel az Advanced File and Print Server a Microsoft NT hálózati technológiáját használja, ezért peer-to-peer kompatibilis azzal, így a kliensek számára pontosan úgy látszik, ahogy egy 4.0-s NT szerver. A rendszer mind a felhasználó, mind pedig a rendszergazda szemszögéből pontosan úgy néz ki, mint egy NT környezet.

Nagygép

A nagygépek felé általában ugyanazokat a protokollokat használják, mint a többi unixos gép felé.

DOS

A unixos fájlok és nyomtatók DOS alóli elérhetőségét biztosító termékek nagy többsége NFS protokollt használ, melynek szerver oldali része minden Unixban megtalálható.

Windows

A kisebb hálózatok hasznosíthatják a főbb Unix platformokon futó SCO VisionFS programcsomagot, amely a munkacsoport alapú windowsos integrációt oldja meg. Lehetővé teszi, hogy a Unixra kapcsolódó windowsos munkaállomások egyszerűen jussanak fel a szerverre.

Ez egy nagyteljesítményű szerver alapú fájl- és nyomtatómegosztási megoldás, amelyet kizárólag a szerverre kell telepíteni, és ezzel a hálózaton minden Windows PC hozzáférhet az összes Unix fájlhoz és printerhez. Tehát az NFS-sel ellentétben nem szükséges a szoftvereket minden egyes PC-re telepíteni és konfigurálni.

A VisionFS azt is lehetővé teszi, hogy munkacsoportok akár az Interneten keresztül is megosszák az erőforrásokat, és távoli Unix szervereket is úgy kezeljenek, mintha azok a helyi hálózaton lennének. A termékben lévő profilszerkesztő programmal egyszerűen és gyorsan lehet a hozzáférési jogokat szabályozni.

Az integráltság magasabb fokát képviseli az SCO SQL Retriever, amely a windowsos munkaállomásokon nyújt ODBC és JDBC felületet szinte az összes unixos SQL-alapú adatbáziskezelőhöz.

Egyéb hálózati kliensek

A Unixware7 a szokásos Windows/DOS kliensek mellett — jó Unix-hoz híven — széleskörű támogatást nyújt más eszközökhöz is. Ilyenek például a nagy reményeket keltő NC (Network Computer), vagyis az úgynevezett vékony kliensek, a Java-képes „Web-top” eszközök, a jól bevált karakteres terminálok (VT220 stb.) és más egyéb ASCII perifériák. Az adott alkalmazás igényeinek legjobban megfelelő környezetet lehet mindenhol alkalmazni.

Itt külön kiemelni az új SCO „Tarantella” technológiát, amely egy Java-képes böngészőprogram segítségével (tehát gyakorlatilag semmiféle karbantartást nem igénylő kliens használatával) lehetővé teszi a hozzáférést szinte bármilyen Unix, NT, mainframe, grafikus vagy karakteres alkalmazáshoz, azáltal, hogy az adott alkalmazás felületét röptében „webesíti”.

Hálózati protokollok

A hálózati számítástechnikához természetesen minden lehetőséget megad a Unixware7, többféle protokoll támogatása, a legfrissebb Internet, Java, üzenetkezelő és címtárkezelési szolgáltatásokkal együtt.

Természetesen támogatja a TCP/IP legújabb verzióját, az IPV6-ot és az ehhez kapcsolódó összes jelentősebb szolgáltatást: NFS, NIS, NTP, DNS, DHCP, LDAP és még hosszan sorolhatnánk a 3-4 betűs rövidítéseket. A szabványos SNMP támogatásával a UnixWare természetesen integrálható bármilyen hálózati felügyeleti rendszerbe, mind ügynök, mind pedig felügyeleti állomás funkciót elláthat.

Backup

A kliens-szerver alapú mentőprogramoknak mind a kliens, mind a szerver változata elérhető a Unix gépeken. Így könnyen tudunk Novell, NT, Windows gépekről a unixos szerverre menteni. De az sem gond, ha a unixos fájlokat kell egy NetWare-es gépre elmenteni. Ez különösen a nagy, heterogén hálózatoknál fontos tulajdonság.

Toporczy István

Weblapjárás

Én sem tudom, miként sokan mások, hogy mi és hol található velem kapcsolatban, bár szeretném tudni. (Vagy jobb mégsem tudni? Na, ezt ki tudja?!)

Tavaly nyártól a mai napig heves érdeklődést tanúsítottam egy igen gazdag tematikájú és gyakorta meg-megújuló weboldal iránt: <http://www.alaplap.hu>, de ma este váratlanul és véletlenül találkoztam az IGAZIVAL (<http://www.hir.hu/alaplap/cikk.htm>), vagy van még ennél is igazibb, csak én kelekótya nem tudok róla, hogy Ő is engem keres, bár nem tud rólam. Talán csak nézem (és olvasom) az új ALAPLAP-ot, de nem látom?

Üdvözet és majdnem gratuláció, mert az újság (és abból szinte mindegyiket ismerem) nagyon régóta még mindig a legjobb.

Kárpáti Ferenc

Ui.: Kicsit bánt, hogy Herczeg úr cikkében (és a NET-en) nem találok a ConfigSYS Kft, illetve a Titkár program elérhetőségét.

Mi sem tudjuk, hogy velünk kapcsolatban mi és hol található a Weben, és saját esetünk is azt példázza, hogy jobb az óvatosság.

Megnyitni egy weboldalt „hirtelen felindulásból” vagy alkalmi jelleggel is lehet. Saját állandó weblappal jelen lenni a Hálón, az már egy kis macerával és felelősséggel jár. Egy nagyobb webhely (web site) fenntartása pedig egyenesen vállalkozás. Weblapunkra irányuló hivatkozásokat, linkeket más weblapokon csak úgy (volna) szabad elhelyezni, ha hosszú távra tudunk tervezni és a szálakat szükség esetén el is tudjuk vágni.

Egy jól komponált weblap elkészítése annyira a „levegőben lógó” kihívás, hogy sokan vállalják akár lelkesedésből is. A weblapot folyamatosan frissen tartani, adatokkal feltölteni, időnként megújítani — az viszont már (sokszor monoton) munka, amit fizetés nélkül kevesen tesznek meg. Ennek anyagi fedezetére azonban az esetek többségében külső forrást kell biztosítani, mert azt magán a Weben kitermelni — kevés kivételtől eltekintve — egyelőre illúzió.

Első nekifutásból mi is szerettünk volna a Weben tartalmilag valami többletet adni a nyomtatott laphoz képest, és ezzel a szándékkal jött létre (vendégként a Budapest Online-nál) a <http://www.hir.hu/alaplap/cikk.htm>. Annak frissítése azonban a lelkesedés fogytával (1996 júniusában) abbamaradt, és azóta ez az oldal internetes „kövületté” vált.

A Telnetnél 1997 nyarán indítottuk a <http://www.alaplap.hu> saját weblapot, amelynek viszont már más funkciót szántuk: azt, hogy bárki számára bármikor hozzáférhetőek legyenek címadataink, hirdetési tarifáink, tartalomjegyzékeink... Tudatosan vállalt „vállalati faliújságunknak” nem is csináltunk olyan propagandát, hogy ennél többet várjanak el tőle. Mindez persze nem jelenti azt, hogy később sem fogunk rajta néhány dolgot kibővíteni vagy megváltoztatni... De azt majd mindig reklámozzuk.

Utóiratával kapcsolatban ezúton kérünk elnézést mindazoktól, akik nem vették elég hamar észre, hogy áprilisi számunkban, mely éppen április 1-jén került forgalomba, „őhercegsége” áprilisi tréfaként írta bele cikkébe a nem létező ConfigSYS Kft-t és a Titkár programot.

Arról viszont sajnos az ÉLET tehet, hogy mindaz, amit abszurditásnak kellett volna érezni, annyira életszerűen hangzott!

(A szerk.)

Halál a régire?!

Pár nappal korábbi faxomat követően rekordgyorsasággal megérkezett a LAP (az alap). A hónap témája és egész tematikája igencsak meglepedésemre szolgált. Ám minden örömből vegyül egy kis ürm is.

A driverek könyvtárában régóta várt anyagok vannak, ám a listából továbbra is roppantul hiányzik nekem egy őskövület, az 1985-ös Phoenix BIOS, amelyet még lemezzel kellett megejtetni az akkor korszerű 286-ossal. Ebben az ügyben már beszéltem szerkesztőségükkel, és annak alapján erre a CD-re vártam — ÉRTIK: ALAPján :-)). Ha lehetne tenni valamit az ügy érdekében, az nem lenne rossz. A 286-os egy kiszuperált jószág, de kísérletezésre és hobby-célokra még igen jól használható lenne.

A CD-hez az OS/2 Team igazán kiváló kezelőfelületet készített. Szilárd elhatározásom, hogy az általuk jegyzett lapra, a Net.Times-ra is elő fogok fizetni, az eddigi lapszámonkénti vásárlás helyett.

Katona Sándor

A régi rendszerek halálra ítétele sajnos a számítástechnikai ipar érdekeltségének és erőviszonyainak egyenes következménye. „A FEJLŐDÉS” két „retrográd” ember-típus létjogosultságát nem hajlandó elfogadni:

1. Azokat, akik megtartják régi, elavult gépüket, és továbbra is azon akarnak dolgozni, játszani, sőt új programokat is futtani.

2. Azokat, akik beszerezik a legkorszerűbb (értsd legújabb) gépkonfigurációt, és azon 'horribile dictu' régi, megszokott programjaikat szeretnék használni, mert azok az adott célnak tökéletesen megfelelnek (néha jobban, mint az újak).

E két renitens embercsoport megértést tanúsít a technikai haladás törvényszerűségei iránt, tisztában is van a visszafelé irányuló kompatibilitás korlátaival, ezért nem kíván lehetetlent — csak lehetőséget. Ugyanakkor a bulldózeres tereprendezésű fejlődés hívei a múlt megőrzésének lehetséges megoldásaitól is elzárkóznak, és saját maguk ilyen célokért semmit nem hajlandók áldozni. A konfliktus pillanatnyilag feloldhatatlannak látszik, intézményesen az ipar „nem támogatja a túlhaladott technika konzerválását”.

Ugyanakkor a számítástechnikusok között vannak olyanok, akik lelkiismeretét mégiscsak birizgálja ez a helyzet, és saját szűkös lehetőségeik között igyekeznek segítséget nyújtani. Ha módjukban áll, megírnak vagy megíratnak a régi eszközök használhatóságát az új körülmények között is fenntartó segédprogramokat, azok elérhetőségéről tájékoztatást adnak, ez ügyben az Interneten másokkal együttműködnek stb. Partizánháborújuk azonban nem sok eredményt ígér, mert egyrészt nincsenek hatással a hardverfejlesztésre, másrészt a nagy szoftverrendszerek forráskódjaihoz jogi vagy fizikai okokból nem férnek hozzá, tehát a megoldások elkészítéséhez nincs meg a szükséges háttérjük. A szoftvercégek pedig általában holtta nyilvánítják ugyan régi programjaikat, de azok forráskódját nem teszik nyilvánossá és szabaddá. Ők már soha sem fogják hasznosítani, de azt sem engedik meg, hogy mások használhassák azokat (lásd: a kertész kutyája.) Olvasónk másik megjegyzéséhez annyit, hogy sajnos a Net.Times további megjelentetése jelen pillanatban bizonytalannak látszik. Következő számunkban talán már beszámolhatunk róla, hogy lesz-e folytatás.

(A szerk.)

Photoshop sulis — VII.

Betűfelület, torzítás, háttér

Nyolcadik lecke

Képes szöveg

Olyan szöveget is készíthetünk, amelynek betűi mintha képekből lennének kivágva. Az eljárás nem is olyan bonyolult, mint elsőre gondolnánk.

— Nyissunk meg egy képfájlt, amelyet a betűkben szeretnénk látni.

— Jelöljük ki az egész képet (Ctrl+A), és másoljuk fel a vágólapra.

— Az Edit legördülő menüből adjuk ki a Define Pattern (minta meghatározása) parancsot.

— Most a File legördülő menüből vagy a Ctrl+N billentyűvel hozzunk létre egy új képet.

— Ezen az új képen helyezzük el a szöveget a Type eszköz segítségével, de a kiválasztást még ne szüntessük meg.

— Az Image/Effects/Scale parancssal nagyítsuk a szöveget a kívánt méretűre, majd hagyjuk jóvá a nagyítást úgy, hogy a nagyítás keretén belülre kattintunk. Ekkor még megmarad a szöveg körül a lebegő kiválasztás.

— Most az Edit legördülő menüből adjuk ki a Fill parancsot, a párbeszédablakban válasszuk a Pattern opciót a

Use legördülőből, és válasszuk ki a festés módját is, ami legyen mondjuk a Darken.

— Az OK gomb megnyomására a kiválasztott szövegbe betöltődik a mintaként definiált kép.

— Szüntessük meg a szöveg körül a kiválasztást. (1. kép.)

Természetesen arra is van mód, hogy egy már létező képre helyezzünk el ilyen feliratot. Ilyen esetben bonyolítja a dolgot, ha a mintaként definiált kép és a célkép nem azonos méretű, illetve más felbontású. (A példában azért küldtük a mintaképet a vágólapra, mert így az újonnan létrehozott kép automatikusan a vágólapon szereplő kép paramétereit veszi fel, ha közben nem változtatunk a paramétereken.) Mindig próbáljuk meg akkora méretben létrehozni a szöveget, amekkorában azt valószínűleg szeretnénk, mert az utólagos méretváltoztatás rontja a betűk kontúrjait.

Ha a Type eszköz ablakába nem tudunk szöveget beírni, akkor ez azért van, mert a célkép paraméterei nem teszik lehetővé a megadott méretű betűk beírását. Ilyenkor csökkentenünk kell a betűméretet, és a szöveget majd utólag kell a megfelelő méretűre nagyítanunk. Ez minden más esetre is vonatkozik a

Type eszköz használatakor! Ügyeljünk arra is, hogy mielőtt a Type eszközre váltanánk, az előtér színét állítsuk tiszta fehérre — hacsak nincs olyan szándékunk, hogy a szövegben a minta keveredjen valamilyen más előtérszínnel.

Mintaként nemcsak egész képet, hanem abból egy részletet is választhatunk, és a kiválasztott részletet a minta definiálása előtt manipulálhatjuk is. Ha a szöveget valamilyen módon torzítani szeretnénk, akkor ezt még a mintával való kitöltés előtt tegyük meg, ellenkező esetben a szövegbe betöltött minta is követni fogja a torzítást. Igen érdekes feliratokat hozhatunk létre, ha annak egy-egy részét (akár minden betűjét) más-más képekkel vagy részletekkel töltjük ki. Ilyenkor viszont a szöveget részenként, illetve betűnként kell létrehozni, a leckénkben ismertetett módszer ismételt alkalmazásával. A részletek vagy betűk pontos illesztését elősegítendő, használjuk az Info palettát — ennek aljában láthatjuk a kurzor pontos helyzetét az X,Y koordináta-rendszerben.

Kilencedik lecke

Torzítások

Talán a színekkel való manipulálásnál is érdekesebb, izgalmasabb feladat az alakok megváltoztatása; a Photoshop erre is kínál lehetőségeket. Alakok, képrészletek torzítására sor kerülhet akár tréfából, akár bizonyos hatások elérése érdekében. Az alább közölt példák egy kis ízelítőt adnak a torzító eszközök használatával készíthető beavatkozásokról. Kiindulóképünk egy portré, amelyet a korábbi példákban is felhasználtunk. (2. kép.)

A 3. kép úgy készült, hogy először kijelöltük az egyik szemet a Lasszó eszközzel, majd kiadtuk az Image/Effects/Distort parancsot. Ugyanezt tettük a másik szemmel is, majd az orral.

A 4. képen ugyanezt tettük az arc alsó felével, majd a középső tincset jelöltük ki, és azt szintén a Distort parancssal lehúztuk. Hogy a hatás természetesebbnek tűnjék, a lehúzott tincset kissé eltérően rajzoltuk körül a Lasszó eszközzel, majd az Image/Flip/Vertical parancssal vízszinte-



1. kép



2. kép

sen megfordítottuk, utána pedig az előbb manipulált tincs mellé helyeztük.

A torzításra speciális szűrőket is használhatunk. Az 5. és a 6. kép a Filter legördülő menü Distort/Pinch szűrőjével készült, de itt kijelölést nem alkalmaztunk, hanem az egész képet torzítottuk. Az elsőn az Amount csúszkát mínuszra, a másodikon pedig pluszra állítottuk.

A torzított részleteket a Move eszközzel igazíthatjuk pontosan a helyükre. Az Anti Aliased opció bejelölésével a kiválasztások finom átmenettel simulnak bele a képbe. Kiválasztásnál a Feather opcióban megadott mérték szerint a körvonal szögletes részei lekere-

kednek — ez különösen hasznos lehet szabadkézi kiválasztásoknál (Lasszó), mert bizonyos mértékben képes korrigálni a kiválasztás pontatlanságát. Minél nagyobb értéket adunk meg, annál erősebb lesz a kerekítés.

Tipp:

Szabadkézi kiválasztásnál mindig nagyítsuk fel a képet, hogy pontosabban dolgozhassunk, de csak annyira, hogy az ablakban elférjen az a részlet, amelyen dolgozni szeretnénk. Ha torzításnál a kijelölt részt összébb húzzuk, a hiányzó részt a háttérszín, illetve az alatta lévő réteg tölti ki. Ez sajnos a program fogyatékosága, ezért vigyázunk, hogy a torzított részletek az eredeti részleteknél inkább nagyobbak, mint kisebbek legyenek — ellenkező esetben bonyolult és precíz retusálást kell alkalmaznunk, hogy az új képrészlet pontosan beleilljék a környezetébe. A részletek bármely irányú széthúzásánál, nagyításánál számolnunk kell azazal, hogy a megnövelt területeken a program matematikai közelítéssel számítja ki az új pixelek számát, ezért minél nagyobb a méretváltozás, annál inkább érzékelhető a beavatkozás, és ez nem tesz jót a képnek. Minél nagyobb az eredeti kép felbontása, annál kevésbé lesz észrevehető a beavatkozás. Mindenképpen tartózkodjunk a túlzott mértékű torzításoktól!

Házi feladatok:

— Próbáljuk meg magunk létrehozni a fenti torzításokat, az eredményből tanulhatunk, ha a referenciaképekkel összehasonlítjuk.

— Más forrásokból származó képeken is alkalmazzunk hasonló torzítás-



3. kép

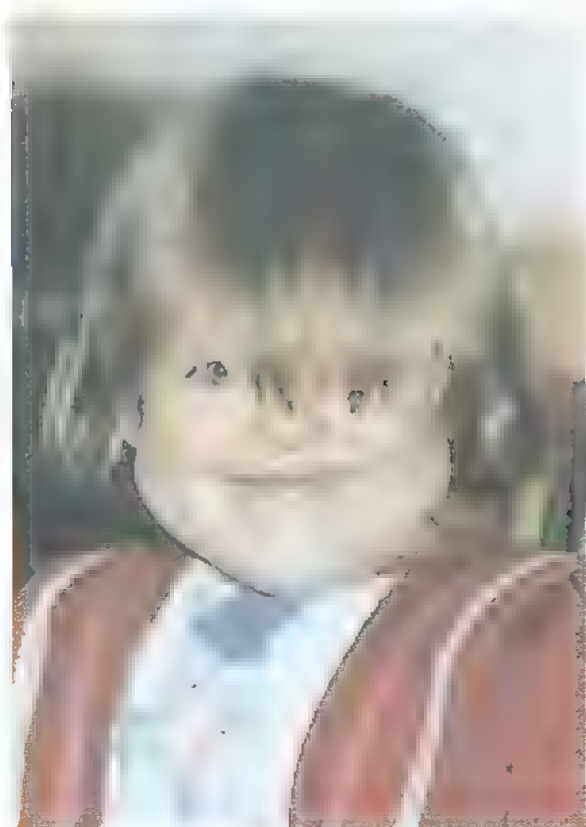
kat, és ezek eredményeit is hasonlítsuk össze a referenciaképekkel.

Egyéni ötletek kipróbálásával kellő gyakorlatra tehetünk szert, és az ilyen irányú gyakorlatok segítenek meglátni azt, hogy egy bizonyos képet hogyan tudunk manipulálni.

Tizedik lecke

Hátterek

Bizonyos feladatoknál megkívántatik, hogy képünket valamilyen háttérrel lássuk el, és arra helyezzünk el feliratokat, kisebb képeket stb. Készíthetünk szép, ízléses reklámokat, címfeliratokat, weblapokat stb., amelyeken a háttér



4. kép



5. kép



6. kép



7. kép

valamilyen anyag illúzióját kelti. Lehet az például fa, márvány, homok, beton, tűz vagy víz, égbolt és szinte bármi más. Lényeg, hogy a háttér ne „üsse” a rajta elhelyezett feliratokat, képelemeket, hanem mind színével, mind pedig mintázatával inkább kiemelje azokat. (Általában nem szerencsés mondjuk víz háttérrel tenni kék betűk mögé, a létrejövő effektust azonban mindig egyedileg kell mérlegelni, nem szabad teljesen az „aranyigazságokra” építeni.)

Ha rendelkezünk olyan képekkel, amelyek ilyen felületeket ábrázolnak, vagy vannak olyan képeink, amelyeken ilyen részletek megfelelő minőségben fellelhetők, akkor a részletet vagy a teljes képet mintaként definiálhatjuk, és ezzel a mintával kitölthetünk egy háttérnek szánt üres képet. (Hasonló feladatot oldottunk meg, amikor betűket töltöttünk ki képekkel, illetve képrészletekkel, lásd nyolcadik lecke.)

— Nyissuk meg a háttérként választott képet, és jelöljük ki annak egészét (Ctrl+A) vagy egy megfelelő részét (négyszögletes kiválasztó eszköz).

— Adjuk ki az Edit\Define Pattern parancsot.

— Hozzunk létre egy új képet a kívánt paraméterekkel.

— Erre a képre adjuk ki az Edit\Fill parancsot, és a párbeszédablakban a Use legördülő listából válasszuk a Pat-



8. kép

tern. A Mode legördülő menüből válasszunk festési módot (például Darken).

— Az OK gomb megnyomására az üres kép kitöltődik a mintaként definiált képpel (képrészlettel).

Nézzük meg ezt képekben is. (7. kép és 8. kép.) Láthatjuk, hogy az új képet a minta folyamatosan tölti ki, mintha egy nagyobb felfelületről készült volna. Esetünkben ez annak köszönhető, hogy a mintaként definiált kép mintázata kifejezetten alkalmas ismétlésre. Más képek esetében, ha a célkép nagyobb, mint a mintául szolgáló, akkor a minta ismétlődése valószínűleg nem lesz ennyire folyamatos. Ilyen esetekben az ismétlődő minták illesztési hibáit a pecsétnyomóval vagy a maszatoló eszköz segítségével kijavíthatjuk, illetve elmoshatjuk.

Tipp:

— Érdekes mintául szolgálhat egy gyűrött papír is, amelyet beszkennelünk. Lehet az fehér is, amelyet azután utólag vagy úgy hagyunk, vagy színezzük, de használhatunk színes (vagy mintás) papírt is.

— Egy üres képen az Add Noise filter/Monochromatic többszöri alkal-

mazásával, és utána az Emboss filterrel beton hatású felületet hozhatunk létre (9. kép). Ezt a betonfelületet ugyancsak színezzhetjük.

— Ha fémszínű háttérre van szükségünk, sima (vagy gyűrött) alufóliát szkenneljünk be. A fényes tárgyak szkennelése azonban rendkívül nehéz feladat, mert nemkívánatos elszíneződések lépnek fel, ezért szkenneléskor jelentősen csökkenteni kell a fényerőt. Ha színes fémfelületet szeretnénk, színes fémfóliákat kell beszkennelnünk, de ezüstfóliát utólag is átszínezzhetünk. A fémfóliák gyakori szkennelése viszont a szkennerek fényérzékelő elemeinek idő előtti elöregedését okozhatja, ezért inkább mentsük el az ilyen képeket, és a későbbiekben használjuk fel azokat ismét.

— Felhős ég háttérrel úgy hozhatunk létre, hogy például fehér háttérszínt és kék előtér-színt határozzunk meg, majd kiadjuk a Filter/Render/Clouds parancsot. (10. kép.)

— Lángtengert szimulálhatunk, ha a fehér háttérszínt narancsvörös előtér-színnel kombináljuk, és az előbb említett szűrőt alkalmazzuk.

Mózes István Miklós



9. kép



10. kép

Junior kontra Shredder

Tanulságos „gépi páros”

Polgár László ötlete volt, hogy lányainak, Juditnak és Zsófiának a világbajnok Anatolij Karpovval, illetve Benkő Pál magyar-amerikai nagymesterrel folytatott rapid-mérkőzésével egyidejűleg két kiváló sakkprogram is megmérkőzzék egymással. A lányok szenzációs győzelme azóta bejárta a világsajtót, de kevesebbet szoltak a tavaly világbajnokságot nyert izraeli Junior és az előző évi világbajnok, a német Shredder érdekfeszítő találkozójáról. Most ezt pótoljuk.

A júniusban lezajlott tornán már nem azok a programok futottak a számítógépen, amelyeknek alkotói a világbajnoki címet köszönheték. A számítógépes sakkban is gyors a haladás, egy-egy verzió a kereskedelmi forgalomban ritkán él tovább egy esztendőnél. A Junior egyik alkotója, Shay Bushinsky az utolsó pillanatban küldte meg egy lemezen 4.6 verziószámú programjának legújabb javítását; Stefan Meyer-Kahlen 1997. évi világső Shredder 1.0 jelű programja helyett pedig a közben elkészült 2.0 verzió állt csatasorba. A tornán lejátszott nyolc játszma közül legalább öt volt kiváló teljesítmény.

A Junior által aratott 5,5:2,5 arányú győzelem körülményei azonban nem voltak teljesen reálisak. Bushinsky azt kérte, hogy a programok 128 MB RAM memóriával felszerelt számítógépen

fussanak. Ehhez az AlphaNet Kft a Digital AMG K6 típusú, 233 MHz-es processzorral működő két gépét adta kölcsön, és Csongrádi József vállalta a szakmai közreműködést. A másik programozó, Meyer-Kahlen (telefonon és Interneten) hozzájárult a szokatlanul nagy memóriájú gépen történő futtatáshoz, de kérte, hogy a többi paraméter értékét is ennek megfelelően növeljék: a napjainkban általánosan használatos 32 MB RAM mellett optimális 12 MB-nyi pozíciótáblázatok (hash tables) és 4 MB-os értékelő táblázatok (evaluation tables) méretét 48, illetve 32 MB-ra. A program ezt lehetővé tette.

A pozíciótáblázatok révén a parti során ismétlődő hadállásokat nem kell mindig újból értékelni, és ez — főként, amikor már kevés báb van játékban — lényegesen gyorsítja a program működését.

Az értékelő táblázatoktól viszont a program szelektivitása függ, vagyis az a képesség, amellyel az eleve rossz változatokat ki tudja szűrni és ezzel tehermentesíti az elemzést. Minél több variáns vizsgálatát hagyja ki ugyanis a program, annál nagyobb lépésmélységig képes a többit értékelni, tehát annál nagyobb az esélye annak, hogy a legjobb lépést választja.

A programozó javaslata, hogy négyszer akkora RAM mellett a további paramétereket is megfelelő mértékben növeljék, logikusnak tűnt, a gép teljesítménye kezdetben mégis igen gyenge volt, főleg az első négy partiban. A Shredder a negyedikben volt talán a leggyengébb, és leginkább ebben mutatkozott meg, hogy miért nem vált be a fenti logika. A rossz lépésekből néha többet lehet tanulni, mint a legszebb játszmákból.

Junior 4.6 — Shredder 2.0

Párosmérkőzés, 4. játszma

1. Hf3 c5 2. e4 Hc6 3. d4 cxd4 4. Hxd4 Hf6 5. Hc3 d6 6. Fg5 e6 7. Vd2 a6 8. 0-0-0 h6 9. Fe3 Fe7 10. f3 Hxd4 11. Fxd4 b5 12. Kb1 e5 13. Ff2 Fe6 14. h4 0-0? (1. ábra.)

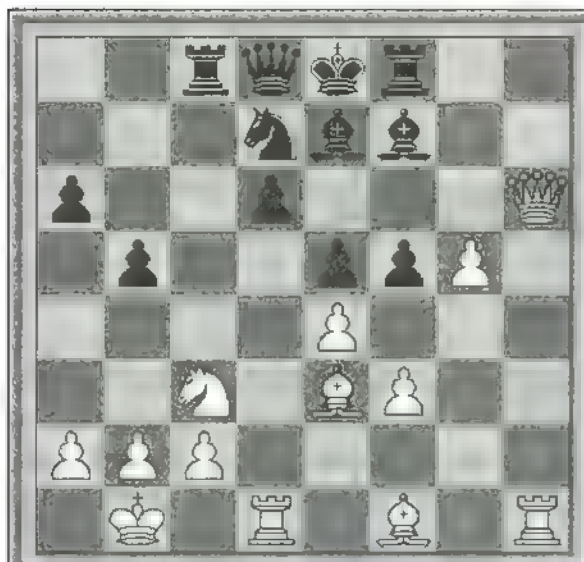
Ha a szicíliai védelemben a két fél ellenkező oldalra sáncol, mindkettő az ellenfél királyállását támadja, ezért éles játék alakul ki. Itt azonban sötét h6-tal meggyengítette saját királyszárnyát, és nem tudta megakadályozni a h vonal megnyitását. Mivel már a vezérolalon is akcióba kezdett, legjobb lett volna királyával középen maradni. 15. g4 Bc8 16. g5 hxg5 17. hxg5 Hd7 18. Fe3 f5 19. Vh2 Kf7 20. Vh5+ g6 21. Vh7+ Ke8 22. Vxg6+ Ff7 23. Vh6 (2. ábra.)

Az elhibázott megnyitás után a sötét király költséges búvóhelyre lelt, csupa tiszt veszi körül. 23. — f4-re 24.g6! fxe3 25. gxf7+ Bxf7 26. Vxe3 következnek, ami után világos gyalog előnye megmarad, és a sötét király ismét kritikus helyzetbe kerül. Ezért minőségáldozattal ellentámadásba fog, ami helyes döntés: 23. — Bxc3! 24. bxc3 Va5 25. exf5 Vxc3? De ez megint rossz választás. 25. — Fxa2+ 26 Kb1 Vxc3 után a világos király nehezebb helyzetbe kerül. 26. Bd3 (3. ábra.)

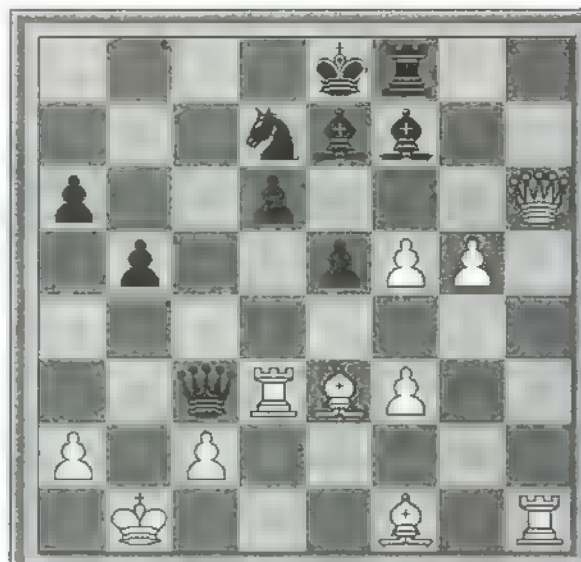
26. — Ve1+?? Kiengedi a döntetlent. 26. — Fxa2+ 27. Kxa2 Vxc2+ 28. Ka1



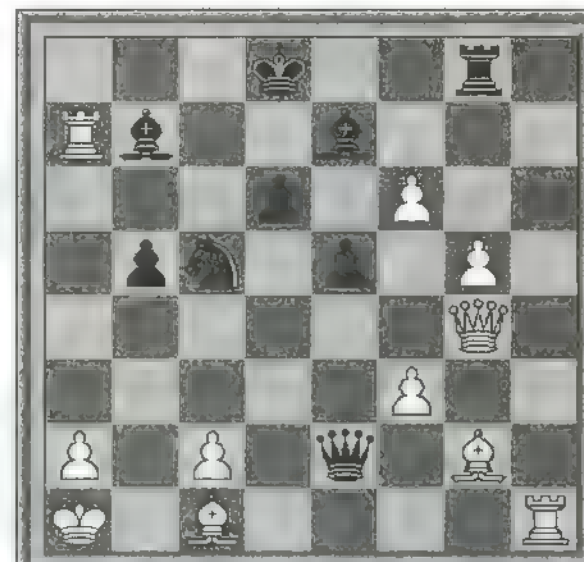
1. ábra



2. ábra



3. ábra



4. ábra

(28. Ka3? d5+ és matt két lépésben.)
28. — Hc5! 29. f6! Hb3+ 30. Bxb3
Vxb3 31. fxe7 Va3+ 32. Kb1 Vb3+ 33.
Kc3 Vxe3+ stb. örökös sakk. 27. Fc1
Vb4+ 28. Ka1 Fc4? Hibák sorozata.
28. — Vc4! — fenyeget Vxa2# — 29.
Kb2 Vxa2+ 30. Kc3 Va5+ 31. Kb2
Va2+; vagy 29. Bb3 Vxc2 még mindig
esélyt nyújtott a döntetlenre. 29. Vh5+
Kd8 30. Vg4 Bg8 31. Ba3! Ve1 32.
Fg2 Ve2 Világos itt és a következő
lépésekben olyan spontán fenyegeté-
sekkel operál, mintha csupán egy-két
lépést számítana ki előre. De már tény-
leg nincs sok lehetősége átfogó terv
építésére. 33. Kb1 Fd5 34. Bxa6 Hc5
35. Ba7 Fe4 Két lépésben matt fenye-
get, de ezt is könnyű kivédeni. 36. Ka1!
Fb7 36. — Vxc2-re nyugodtan üthető
a futó; 36. — Fxc2-re pedig 37. Bxe7!
Kxe7 38. f6+ után az összekötött szabad
gyalogok döntenek. A szöveglépést fi-
nom kettős áldozat követi. 37. f6!
Visszafordul a kocka. (4. ábra.)

A következő bonyodalmakból világ-
gos tisztelónnyel kerül ki. 37. — Fxf6
38. Bxb7 Hxb7 39. Ve6 Sötét hadállása
szétesett. 39. — Vc4 40. Vxf6+ Kc7 A
többi már csak technika dolga. 41. Ve7+
Kb8 42. f4! Vc8 43. Fxb7 Vxb7 44.
Vxd6+ Vc7 45. Vd5 Vc4 46. Vxe5+
Kc8 47. f5 és sötét feladta.

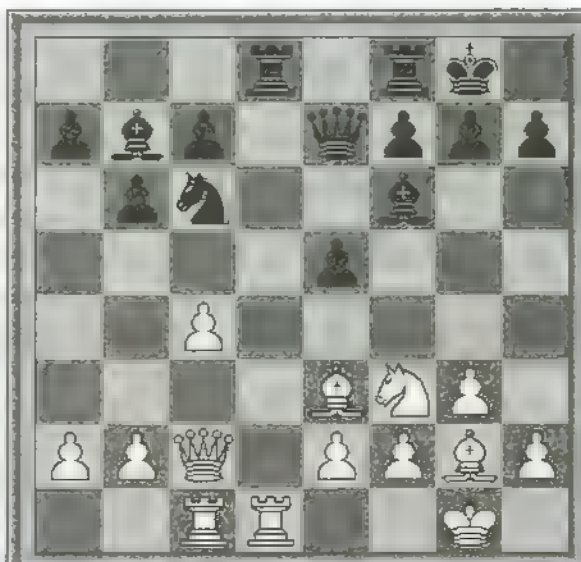
Ebből a játszmából világosan kide-
rült, hogy a Shredder a megnövelt táblá-
zati paraméterekkel nem képes kellő
mélységben számítani. A kezelési uta-
sításban rá is bukkantunk egy figyel-
meztetésre, hogy gyors játék esetén
nem célszerű a pozíció- és értékelő
táblákat növelni, mert a programnak
minden lépésnél időbe kerül a standard
paraméterektől való eltérés, így összes-
ségében többet veszít a réven, mint
amennyit nyer a vámon, és nem marad
elegendő ideje a szükséges mélységű
elemzésre. A spontán ötlet nem vált be,
a gyakorlatban kellett volna kikísérle-
tezni, tesztelni, hogy 128 vagy 64 MB

RAM esetén mennyi lehet a többi pa-
raméter optimális értéke. Az ellenfél
Junior programozója viszont éppen a
nagyobb RAM-kapacitás használatára
készült fel. Amikor visszaállítottuk az
eredeti paramétereket, a mérkőzés má-
sodik felében Shredder bizonyult jobb-
nak, 2,5:1,5 arányban.

Azoknak, akik korábban már olvas-
ták ismertetéseinket a különböző prog-
ramokról, bizonyára feltűnt, hogy az
ábrák a Fritzből valók. Ennek oka a
Junior és a Fritz együttműködése. Min-
den olyan funkciót, amely a sakkozás-
hoz szükséges, a Junior lát el, ugyan-
akkor az adatok prezentációját — az
ábrákat, a lépéseket, az elemzéseket —
a Fritz5 grafikai környezete jeleníti meg
a képernyőn.

Annak illusztrálására, hogy a nagy
mélységű elemzés érdekében mennyire
szükség van a program kellő szelekti-
vítása révén nyert időre, bemutatom a
párosmérkőzés 2. partijának a meg-
nyitás utáni szakaszban előállt, teljesen
ártalmatlannak tűnő hadállást. (5. ábra.)

Még a nagymesterek sem gondolná-
nak itt valami gyors döntésre. A Junior
azonban megtalálta az állásban mélyen
rejlő lehetőséget. Következett: 16. c5!
Alapjában véve kézenfekvő előretörés,



5. ábra. Hadállás sötét 15. lépése után

és csaknem hihetetlen, hogy milyen
hosszú számítás előzte meg. Bxd1+
17. Bxd1 Bd8 18. Bxd8+ Vxd8 19.
Hg5!! E szép lépést Junior igen gyorsan
húzta meg, elárulta vele, hogy követ-
kezményeit már a 16. lépésnél kiszámít-
totta. 19. — Fxg5 20. Fxg5 Vxg5
21. cxb6! Sötét elveszett, mert ha meg-
próbálja tartani tisztelónnyét, mattot kap:
21. — Ha5? 22. Fxb7 Hxb7 23. bxc7
Hd6 24. c8V+ Hxc8 25. Vxc8+ és matt
a következő lépésben. Sötét másik le-
hetősége: 21. — Vg6, ami átmenetileg
védi a c6 huszárt, de erre 22. Vxg6 hxc6
23. bxc7! Ha5 24. Fxb7 Hxb7 25. c8V+;
vagy 23. — He7 24. Fxb7 majd 25.
c8V+ Hxc8 26. Fxc8 nyer. A választott
folytatás sem jobb, mint ezek. 21. —
Hd4 22. Vxc7 Hxe2+ 23. Kf1 Fxg2+
24. Kxg2 Hf4+ 25. Kf1 g6 Megint
matt fenyegetett. Érdekes variáns: 25.
— He6 26. Vc8+ Vd8 és világos vá-
laszthat 27. b7 majd b8V, vagy 27.
Vxd8+ Hxd8 28. bxa7 majd a8V között.
26. b7 Hd3 27. b8V + Kg7 28. Vc3 és
sötét feladta.

A szép parti befejezése után végig-
gondoltam, mik lehettek világos számí-
tásai, amikor 16. c5!-öt meghúzta. Szá-
molt a kézenfekvő kettős bástyacseré-
vel, és utána — talán már nem elsőként
— a 19. lépés tisztázódásával, amelyet
szinte fonalasan követtek a kölcsönös
ütések, ezeknek alig volt alternatívája.
Kiderült továbbá, hogy a két főváltozat
közül az első a bástyák tervezett cseré-
jével induló és mattal végződő, a má-
sodik (többféleképpen is) a gyalog át-
változására vezető lebonyolítással, egy-
aránt tizenegy (!) lépéspár előreszámí-
tását igényelte. Azután „megkérdeztük”
a gépet (hiszen az „jegyzőkönyvében”
minden egyes lépésénél regisztrálja a
számítási mélységet), hogy hány lépé-
sig elemzett 16. c5!-nél, és valóban a
11-es szám jelent meg. Ez különleges
teljesítmény.

Lindner László

Delphi 4 — immár az Inprise-től

Az Allegro bemutatkozása

Az Inprise Inc. (a Borland utódja) 1998. június 15-én bejelentette a Delphi új, 4. verzióját. Az Allegro kódnevű Object Pascal fejlesztőeszköz megtartotta a korábbi verziók számos jól bevált tulajdonságát, de több újdonsággal is büszkélkedhet, mind grafikus fejlesztőfelületét (IDE), mind nyelvi képességeit tekintve.

A Delphi 4 fejlesztőeszköz általános képe emlékeztet ugyan a korábbi verziókéra, de mindjárt az indítás után szembeötlő az eszközsorok különbsége. Itt a hasonlóság a Win32 rendszerek több más programjában megszokott eszközsorokkal már jelzi, hogy ezek az ablakelemek szabadon áthelyezhetők. Akár a főmenüt is lebegőablakká varázsolhatjuk, és arrébb tehetjük. De lehetőség van ezeknek az ablakoknak a dokkolására is a szerkesztőablak vagy az Object Inspector keretébe.

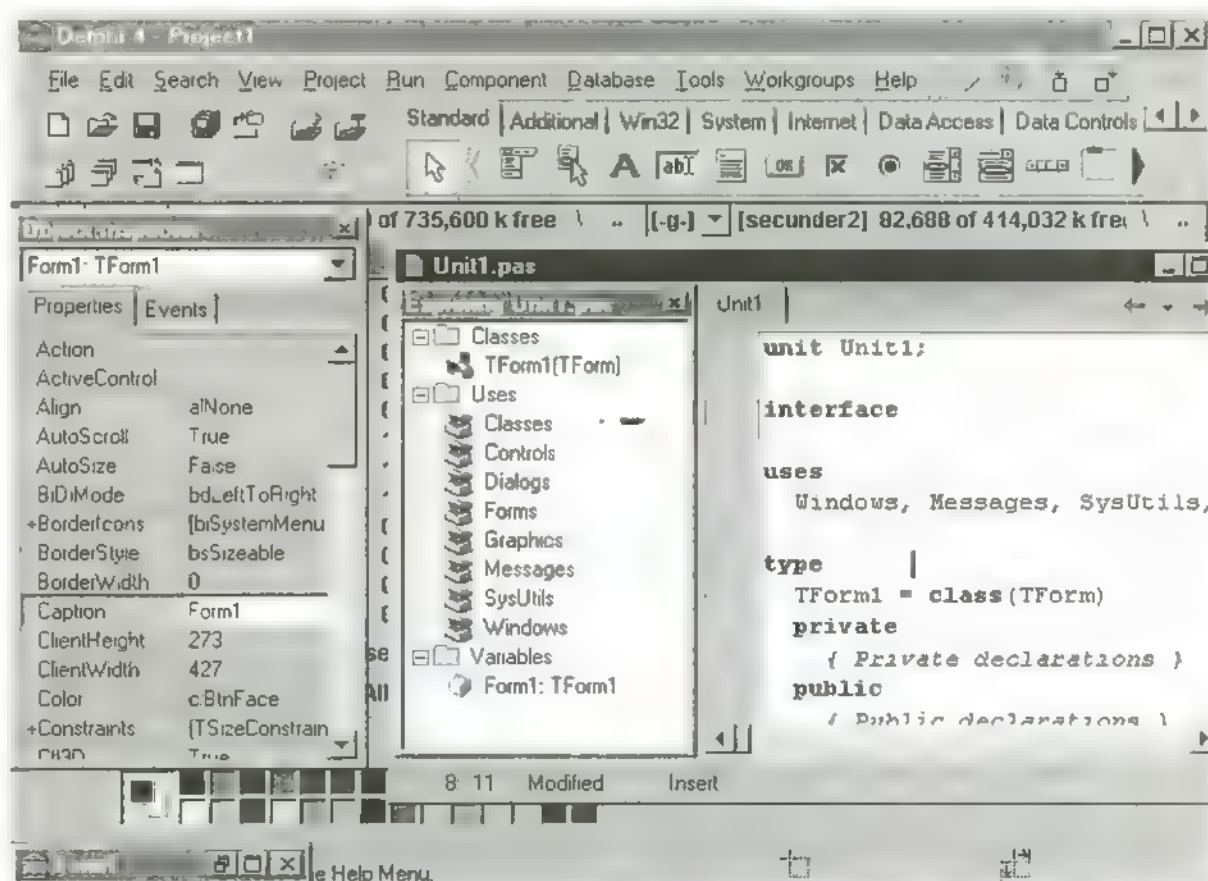
Ez utóbbi lehetőségnek azonban nem annyira a menüablakoknál vehetjük hasznát, hanem inkább a programok fejlesztése, tesztelése során, a változók figyelésekor, vagy például a futás nyom követésekor. Ezek a korábbi verziókban csak egymás alá és fölé helyezhető ablakok, a Delphi 4-esben már szintén dokkolhatók, így a programfejlesztés során mindig szem előtt vannak, és a program debuggolása során megszűnik az éppen esedékes ablakok keresgélése. Szintén a hibakeresést segíti, hogy immár a Delphiben is lehetőség van a CBUILDER 3-ashoz hasonló módon a CPU-szintű bogarászásra.

Mielőtt azonban a programfejlesztésben a hibakeresésig eljutnánk, számos más segítséget is kapunk. A forráskód-szerkesztő ablak kiegészült egy további elemmel, így folyamatosan figyelemmel kísérhetők a szerkesztett forrásfájl osztályai, változói és eljárásai. Természetesen nemcsak a szemlélődést, hanem a navigálást is megkönnyítik, így többé nem kell egy terjedelmes unitfájl-

ban állandóan a szöveg görgetésével múltatni az időt. Gyorsan áttekinthetjük annak vázlatos tartalmát, és eljuthatunk a szerkesztendő kódsorokig. Ebben a Ctrl+Shift+Felső kurzor nyíl, illetve a Ctrl+Shift+Alsó kurzor nyíl billentyűkombinációk is segítenek, amelyekkel

hadtak a programozók. Ilyen az automatikus kódkiegészítés, amelyben új elem az automatikus osztálykiegészítés. Ennek segítségével már osztályok vázát is többé kevésbé automatikusan lerakhatjuk. Elegendő a unit interface részében a Ctrl+Shift+C billentyűkombinációt megnyomni, és a kódgeneráló szükség szerint automatikusan létrehozza az eljárások vázait, a propertyk író-olvasó eljárásait. Így különösen a komponenseket fejlesztő programozók kapnak jelentős segítséget.

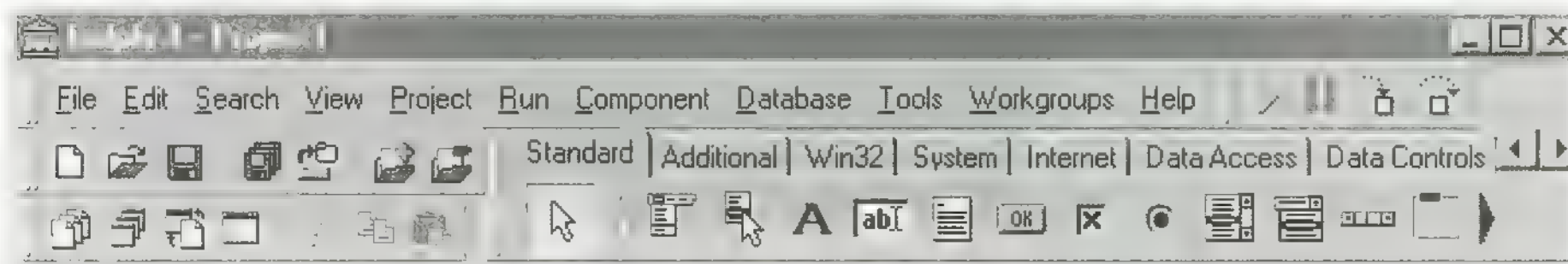
Ahogy valószínűleg szintén a komponensfejlesztőket fogja segíteni az az



egy unit interface és implementation része között navigálhatunk. Innen már csak a programozó felelőssége, hogy egy-egy unit ne váljon e segédeszközök nélkül áttekinthetetlen kódraktárrá.

Néhány új eszközzel egészültek ki azok a fejlesztőeszköz-elemek is, amelyeket a korábbiakban már megszok-

új nyelvi elem is, hogy immár az Object Pascalban is lehetőség nyílik egy osztályban azonos néven több, csak azok paraméterezésében eltérő eljárások definiálására. Akár konstruktorból is lehet azonos néven kettő, eltérő paraméterekkel. Ezt a korábbi verziókban csak eltérő névvel lehetett megvalósítani,



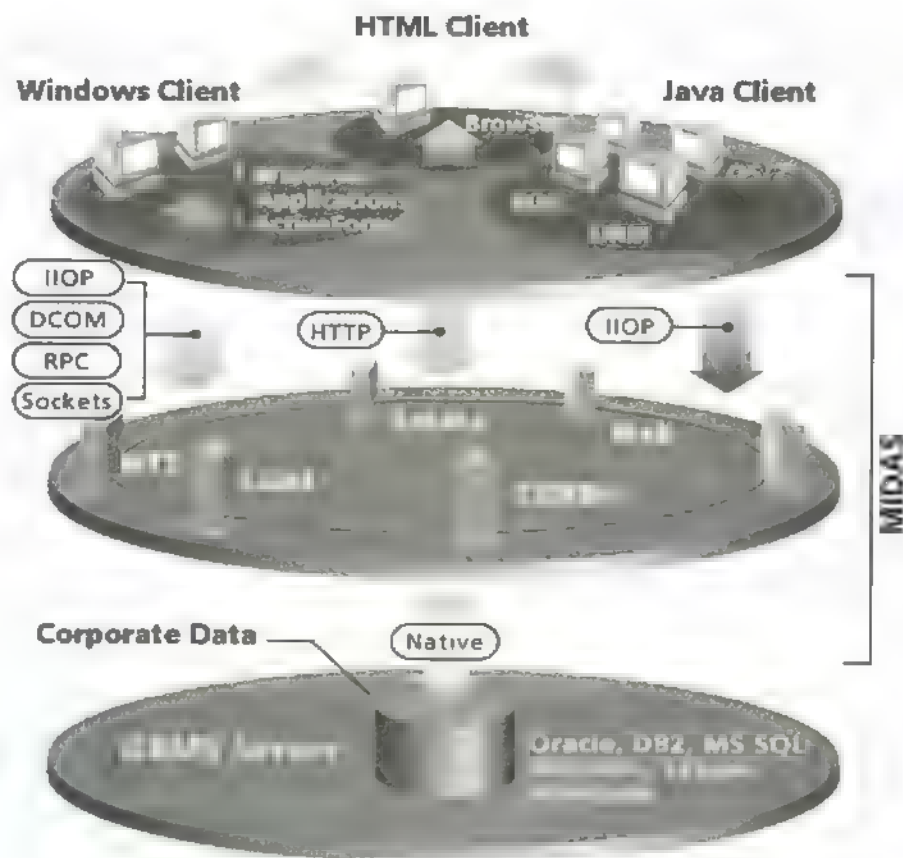
most az új „overload” kulcsszó jelzi a fordítónak a paraméterfüggő hívások generálását. Ezzel a más nyelveken, például Javában programozók kapnak segítséget a Delphivel (is) végzett munkához.

Az említetten kívül azonban több más nyelvi bővülés is jelzi az új Pascal-eszközt. Ezek egyike a dinamikus tömbök deklarálásának lehetősége (A: array of string), amellyel csak a lefoglalandó tömb típusát rögzíthetjük, anélkül, hogy rögzítenünk kellene annak elemszámát is. Szintén jelentős újdonság az alapértelmezett értékek definiálásának lehetősége egy eljárás kódjának megírásakor, például "procedure Valami (B: Boolean; Value: Integer = 0);". Így egyes paraméterek megadása az eljáráshíváskor opcionálissá tehető, például a "Valami (True);" hívás is helyes lesz.

A fejlesztők a C++ nyelvhez való közelítés fokozásán túl valószínűleg már a 64 bites rendszerek felé kívántak átjárást nyitni az új, hosszú változótipusok bevezetésével. Ilyen az egész számok közül a 64 bites egész, az Int64, amellyel nagy negatív és pozitív számok is kezelhetők ($-2^{63}..2^{63}-1$), vagy a pozitív tartományban értelmezett, szintén 64 bites Longword (0..4294967295).

Szintén a számkezelést érinti, hogy a hagyományos 48 bites real típust felváltotta az Intel CPU-hoz jobban illeszkedő, 64 bites double típus.

Ezekkel az elemekkel hatékonyabb programkód megírásának lehetősége nyílt meg a programozók előtt. Azoknak, akik nem kívánnak a programozás ilyen mélységeibe alászállni, a vizuális programeszközök terén is tartogat újdonságot a Delphi 4, ugyanis a VCL (Visual Component Library) szintén bővült. Még az alapobjektum (TObject) is kapott új eljárásokat. Ezekkel jobban programozhatóvá válik az objektum keletkezésének és elmúlásának folyamata (AfterConstruction és BeforeDestruction). De ezentúl nem okoz problémát egy dokkolható ablakokat tartalmazó alkalmazás vagy Windows NT-n egy szervizalkalmazás megírása sem. Az előbbi támogatását a TControl és TWinControl osztályok bővítése, az utóbbit a TServiceApplication és TService osztályok biztosítják.



Az említetteken túl is kibővült a komponenspaletta a korábbi Delphi-verziókhoz képest. A Delphi 4-esbe is belekerültek azok a komponensek, amelyek a CBuilder 3-asban mutatkoztak be. Ezek elsősorban az adatbázis-kezelésben és a intranet-, illetve Internet-alkalmazások készítésében jelentenek bővülést. Például a jelen Delphi-verzió teljes támogatást biztosít a Corba-fejlesztéseknek, és a palettán megtaláljuk a teljes Midas-gyűjteményt is.

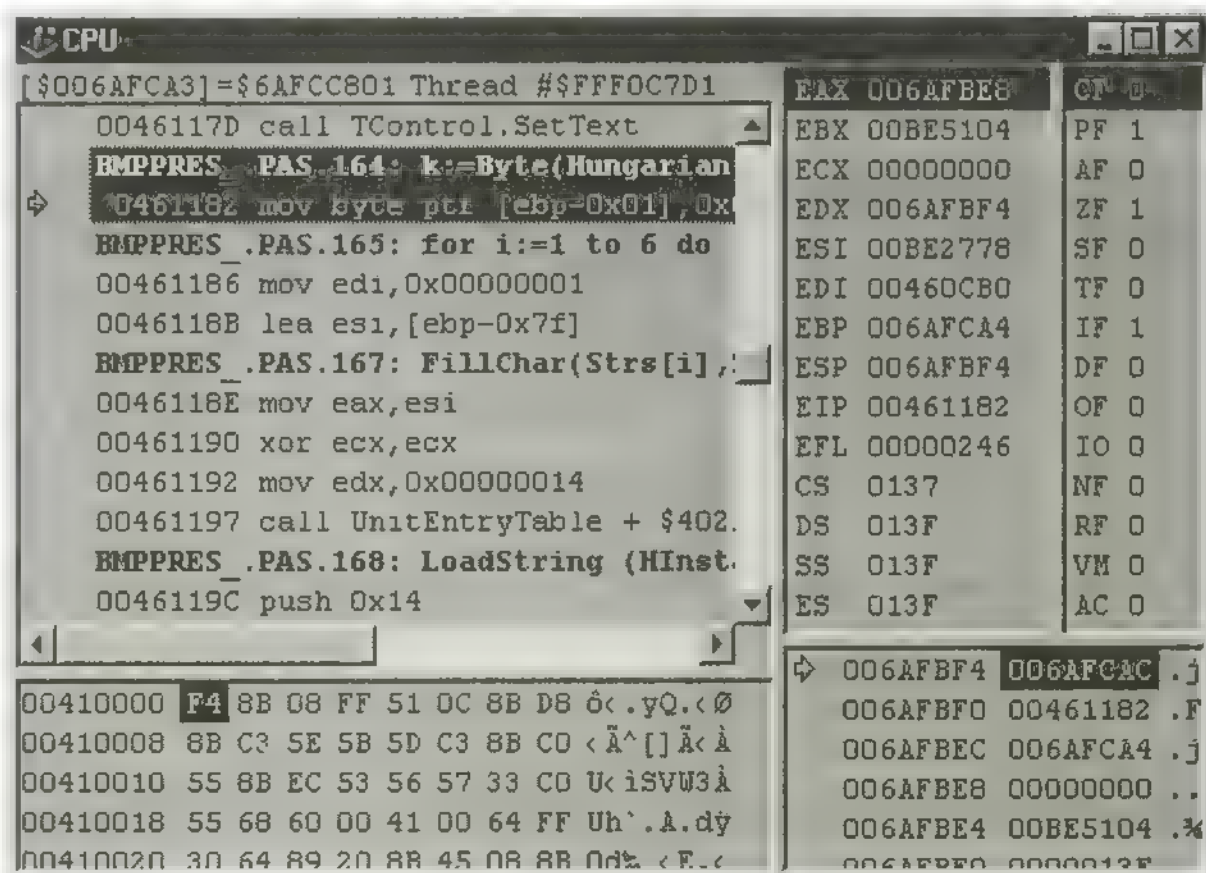
Az elérhető adatbázisok köre is bővült, mivel az új Borland Database Engine (BDE) támogatást tartalmaz az Access 97 és az Oracle 8 felé is, támogatva az Oracle 8 SQL-kiegészítéseket is, mint amilyen az ADT (Abstract Data Types). Az SQL-alapú prog-

ramkészítést egy további eszköz is segíti. Ez a korábbi Visual Query Builder helyett használandó SQL Builder, amellyel hatékonyabban lehet a lekérdezéseket elvégezni. Az újdonságokon túl azonban fontos lehet a régebbi verziókkal való kompatibilitás megőrzése. Elsősorban a forráskód szintjén. Ezt tekintve, és még a 16 bites Delphi 1.02-vel fejlesztett több komponenst, illetve alkalmazást újrarendítve azt tapasztaltam, hogy ha a kód csak Pascal-elemeket tartalmazott, legtöbbször probléma nélkül sikerült a forráskód újrarendítése. Ez a nem egyszer DOS-specifikus assem-

berbetéteket tartalmazó kódokra már értelemszerűen nem volt igaz.

Nem okozott problémát a Delphi 3-as kódok újrarendítése. Azonban a komponenseket és komponenscsomagokat (package) ez esetben is újra kellett fordítani. Az utóbbiaknál azt az eljárást alkalmaztam, hogy a .DPK kiterjesztésű szöveges fájlokban a VC*30.* bejegyzéseket VC*40.*-re javítottam, majd ezeket, illetve a *.PAS forrásfájlokat importáltam a Delphi 4-esbe, elkerülve a csomagok újbóli összeállítását. Ezt követően az alkalmazások fordításával sem volt probléma, de a próbaképpen elvégzett ActiveX-generálással és -felhasználással sem. Ez utóbbi lépések továbbra is támogatottak.

Simay Endre István



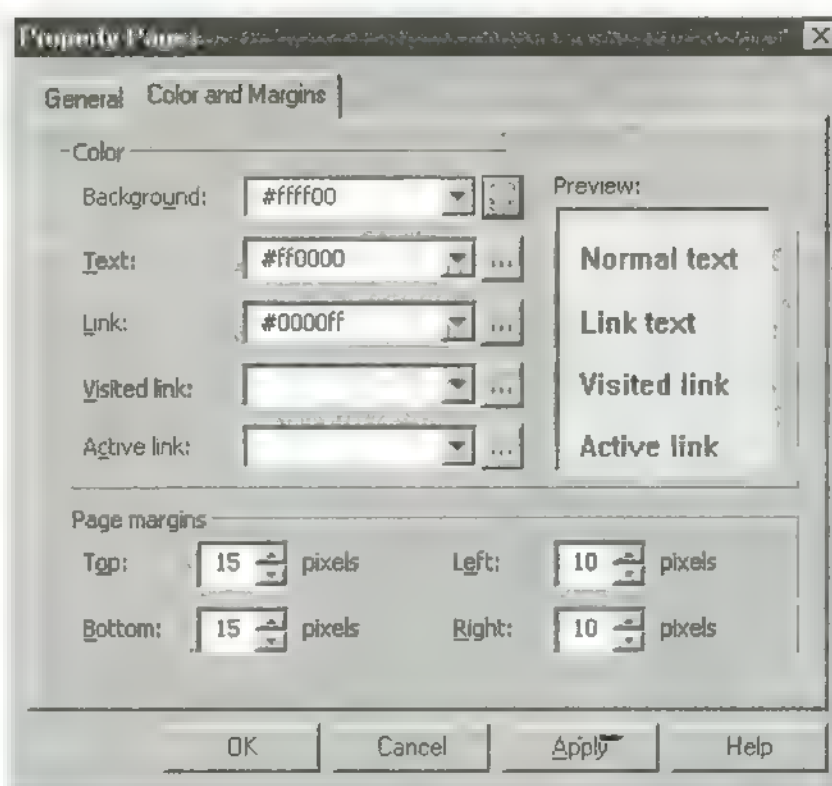
A Studio dinamikus tagja

InterDev a Webhez

Az Internet terjedésével egyre nagyobb az igény olyan alkalmazások iránt, amelyekkel megjelentethetjük adatainkat a Világhálón. Ezek egyike, a Microsoft Visual InterDev, a webes alkalmazásokhoz, fejlesztésekhez ad keretet. A Visual Studio új verziójával ennek a stúdiótagnak is megjelent új, 6.0-s változata. E kézirat lezárásakor még csak béta-változatban.

A Visual Studio, illetve abban a Visual InterDev integrált fejlesztőkörnyezetet biztosít, mellyel kialakíthatók az aktív szerveroldalak (Active Server Pages, ASP), beágyazott JavaScriptekkel, illetve VBScriptekkel együtt. Ehhez járul a WYSIWYG szerkesztési lehetőség a HTML-oldalak készítéséhez, a varázslóalapú integrált segítség az adatbázisok eléréséhez, felhasználásához. Mindezekhez a munkákhoz számos „gyári” komponenst és sablont is felhasználhatunk.

A Visual InterDev 6.0 általam telepített béta-változata a <http://www.microsoft.com/vinterdev> címről is elérhető. Teljes telepítéséhez az ideiglenes állományok elhelyezéséhez a kb. 165 MB-os letölthető anyag mintegy 450-500 MB-nyi merevlemez-területet és természetesen 32 bites Windows operációs rendszert igényel. Telepítéskor üzenetet küldött, hogy a registry szabad tartománya elégtelen a telepítéshez. Ennek ellenére nem okozott problémát a program telepítése és kipróbálása. Szintén igényli az MS Internet Explorer 4.01-es verzióját. Problémát jelenthet azonban az időközben megjelent béta-változat



telepítéskor, hogy azzal nem működik együtt. Ez az utóbbi a program dokumentációjában is olvasható, azzal az ígérettel a Microsoft részéről, hogy a végleges verziók ezen hibáját ki fogják javítani.

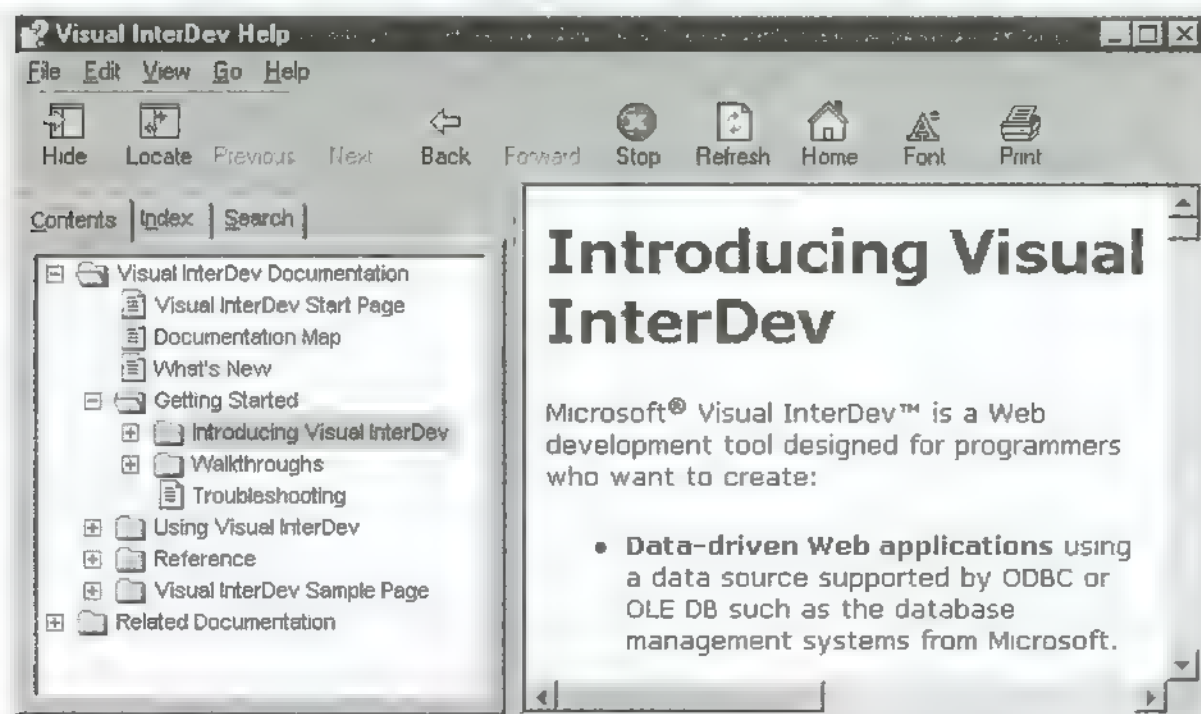
Az InterDev szervesen beépül a Visual Studio adta közös programkeretbe.

Így a későbbiekben a Visual Studio 6.0 közös indulóvarázslója szolgál támpontul mind a korábban telepített programok, mind az InterDev-alkalmazások és projektek fejlesztéséhez. Akár Web-alkalmazásról, akár szerveralkalmazásról legyen szó, a későbbi munkát szintén varázslók segítik. Így gyorsan fejleszthetünk aktív és interaktív Web-oldalakat. Az ASP fejlesztésekor pedig felhasználhatjuk mindazokat a komponenseket, amelyeket például a Visual J++ segítségével is. Beleértve akár más programozási nyelven megírt, akár Windows-rendszerbe installált további COM-komponenseket.

Itt a platformfüggetlenség is kevésbé jelent korlátozást, mivel ezen Web-oldalak aktív részei elsősorban a szerveroldalon futnak. Például egy adatbázislekérdezés lefut a szerveroldali adatbázison, majd dinamikusan, futásidőben felépül a lekérdezés eredményeit tartalmazó HTML-oldal. A felhasználó ezt fogja látni a böngészőjében, anélkül, hogy az adatbázist át kellene töltenie a saját gépére, vagy a lekérdezéssel a munkaállomást kellene terhelni.

A Visual InterDev azonban a kliensoldali fejlesztéshez is biztosítja a fejlesztési keretet. Itt felhasználhatjuk a paletta HTML-komponenseit, melyeknek alapparamétereit a property-szerkesztővel állíthatjuk be. A megjelenítés, és szöveg esetén a menü is segítségünkre van. A HTML-oldalak szerkesztéséhez azonban más programokat is felhasználhatunk, mint például a MS FrontPage vagy az MS WEB Publisher, mely programok szintén támogatják a közös Windows-komponensek felhasználását. Választhatjuk azt a megoldást is, hogy a HTML-oldalak alapjait ezekkel a programokkal rakjuk le, majd a Visual InterDevvel visszük be az aktív elemeket a már kész oldalakba. Ehhez a munkához ad támogatást például a FrontPage '98 szerveroldali kiegészítése. A varázslókon kívül a program használatához a Visual Studióban megszokott MSDN-alapú helprendszer ad támogatást. Ezzel a teljes fejlesztési folyamatban a projektek indításától kapunk átfogó segítséget.

Simay Endre István



GNU for DOS — VI.

Makefile „munkakerülőknek”

GNU-sorozatunkat a terjedelmes anyagok tavaszi torlódása miatt 3 hónapos kihagyással folytatjuk. A mostani rész leginkább a Programozástechnika rovatba illik, mert azt mutatja be a programozóknak, hogy a mechanikus és ismétlődő begépeléstől meg lehet szabadulni egyes segédeszközök és egy kis fantázia segítségével.

A jobb programok forráskódban (is) terjednek. Unix esetén, amikor különféle hardverre szánjuk ugyanazt a programot, ez nem is lehet másképp. Ilyen esetekben a felhasználónak ugyanazokat a fordítási, linkelési feladatokat kell végrehajtaniuk, mint a programozónak...

Valaha a Basic nyelvű és a gépi kódú programokat egy ültő helyben megírtam, azaz amíg a program kész nem lett, és nem működött úgy, ahogy azt elvártam tőle, addig nem is vettem fel kazettára. Amikor áttértem PC-re — és vele együtt a lemezek használatára —, akkor már a félkész programok is helyet kaptak a lemezen. Egy idő múlva méretesebb programokat is írtam, és kényelmi szempontból a forrást több forrásfájlra tördeltem.

Ne dolgozz feleslegesen!

A Borland-környezetek elfedték a fordítási, linkelési feladatokat, így nem volt gond, mit is kell csinálnom egy kis változtatás után, mert minden automatikusan elkészült. Programozás mellett igen sok TeX-ben kiszedett dokumentumot is készítettem. Itt nem volt meg a már megszokott programozói felület, tehát kézzel kellett minden konvertálási, fordítási parancsot kiadnom. Néha többen dolgoztunk ugyanazon a feladaton, és alaposan át kellett gondolnom, hogy mi mindent és milyen sorrendben kell elvégezni. Hogy ne hagyjak ki semmit, a lényegesebb változtatások után a biztonság kedvéért minden parancsot újra végrehajtottam. Erre azért is szükség volt, mert a tartalomjegyzékek, tárgymutatók beillesztése után esetleg megváltozhattak a hivatkozásokhoz tartozó oldalszámok, tehát az indexeket újból és újból el kellett készíteni.

Azóta láttam egyet s mást, s ma már tudom, hogyan csinálják ezt a nagyok!

Megfelelő batchfájlban leírhatnánk az összes ilyen utasítást, és ez meg is

felelne a felhasználónak, mert neki egyszer az összes ilyen parancsot végre kell hajtania. Ugyanez a módszer a programozónak viszont gazdaságtalan lenne, mert ő nemcsak egyszer „készíti el” a programot, hanem akár több tucatszor is, mire az jól működik. Ha például átírja azt a header fájlt, amely a nyolc forrásfájlból csak kettőre van hatással, akkor a nyolc fordítás és egy linkelés helyett elegendő két fordítás és egy linkelés, feltéve, ha a többi object fájl még megvan. Nagyobb programokra ez még inkább érvényes, ezért a programozó számára fontos, hogy csak az okvetlenül szükséges fordításokat, linkeléseket végezze el, ám ezeket a parancsokat tizenhatodjára begépelni már igencsak unalmas dolog, amit jó lenne automatizálni.

Az alapvető kérdés az, hogy a fordítással, linkeléssel kapott fájl létrehozása óta megváltozott-e az annak elkészítésében szerepet játszó valamelyik fájl? Itt most a fájl tartalmát figyelő programokat is lehetne használni, de ezek erőforrásigénye elképesztő. (Emlékezzünk csak a korábban bemutatott diffre és az ezt alkalmazó RCS-re, amely néha leáll DOS alatt, mert elfogyott a memória.) Nagyobb programrendszerek készítésekor nem ritka, hogy több száz fájlt kell vizsgálni, ezért valami egyszerűbbet célszerű alkalmazni.

Miután jó ideje szinte minden számítógép tartalmaz folyton járó belső órát, a fájlok létrehozásának, pontosabban utolsó változtatásának ideje alapján megállapíthatjuk, hogy szükséges-e újra elvégezni a fordítást. Ha a gépek órája jól jár, akkor ez a vizsgálat jó eredményt szolgáltat számunkra. A legprimitívebb automatizálás ennek alapján az lehet, hogy készítünk egy olyan programot, amely megállapítja, hogy az utána felsorolt fájlok közül az utolsó-e a legfrissebb. (Hasonlóképpen készíthető olyan program is, amely az első

fájlt vizsgálja az utolsó helyett, de most nem ez a sorrend a lényeg.)

Tehát, ha előbb felsoroljuk a forrásfájlokat, utánuk pedig a belőlük készített fájl neve áll, ez a program akkor ad vissza 1 értéket, ha valamely forrásfájl megváltozott, így szükséges az újrafordítás. Természetesen, ha az utolsó fájl nem létezik, akkor is szükséges végrehajtani a fordítási parancsot. A Simtel msdos/pgmutil alkönyvtárban fellelhető pcc12c.zip közprogram tartalmaz egy ilyen segédprogramot 'later' néven, és ezzel például így nézne ki a batchfájl egyik részlete:

```
later moda.c mod.h moda.o
if errorlevel 1 pcc moda
if errorlevel 1 goto stop
```

Ezt a batchfájlt futtatva csak a valóban szükséges fordítások hajtódnak végre. Mindenesetre oda kell figyelni a feltételek sorrendjére, mert ha azután változik meg az object fájl dátuma, hogy azt összevetettük a forrásfájl dátumával, akkor a szükséges linkelés elmarad! Nem csupán ezért hátrányos ez a forma, hanem azért is, mert sokat kell gépelni (bár leegyszerűsíthető lenne az awk használatával).

Makfile és make

Miután igen sokan programoznak, a fordítások, linkelések automatizálása mindennapos probléma, ezért erre elég régen elkészült a 'make' nevű program. A make által használt formátum a következő:

```
moda.o: moda.c mod.h
      pcc moda
```

azaz a kettőspont előtt szerepel a célfájl neve, mögötte azon fájlok nevének sorozata, amelyekből a célfájl függ. A következő (TAB karakterrel kezdődő) sorban vagy sorokban azok a parancsok kapnak helyet, amelyekkel a célfájl megkapható. (Más megvalósításokban nem TAB karakterek jelölik a parancssorokat, hanem ezek az első üres sorig tartanak.) E szabályok általában egy 'Makefile' elnevezésű fájlban találhatók, és a szükséges műveletek a 'make' parancs kiadása után hajtódnak végre.

A szabályok leírásának sorrendje itt már jóval szabadabb, mint az előző

esetben, ám az első helyen a legfontosabb célnak, általában a futtatható fájl nevének kell szerepelnie. Indulásakor a make felépíti a fájlok függőségi fáját, és ennek elkészülte után a levelektől (azaz a forrásfájloktól) visszafelé ellenőrzi, hogy szükséges-e valamely parancsot végrehajtania. Mire visszaér a fa gyökerébe, minden szükséges fordítást végrehajtott. Ennyi ismerettel már el is boldogulnánk, hiszen csak le kell írni a szabályokat, és el kell indítani a make-et.

Az ember alapvetően lusta teremtes: mindent kitalál, hogy kevesebbet kelljen dolgoznia. Mivel a futtatható fájl függ az összes object fájlától, így azokat a szabályban is fel kell sorolni, és a linkelési parancsban is. Kétszer ugyanazt a listát, de egy terjedelmesebb lista esetén már könnyű eltéveszteni a gépeletést vagy kihagyni egy nevet, ezért bevezették a változók használatát. Mondjuk leírjuk az alábbi utasítást:

```
objs =
proga.o prog.b.o prog.c.o prog.d.o
```

és akkor a szabályok között majd a

```
prg : $(objs)
      link $(objs)
```

szerepel. Ekkor, ha a listát meg kell változtatni, elegendő a változtatást egy helyen kiadni. Ezzel sok hibalehetőséget kiszűrhetünk. A változó eme hagyományos definiálásának vannak hátrányai (miután minden előfordulásakor kifejtésre kerül), ezért a gnumake ismeri a ':=' definiálást is, és az ilyen változó igencsak hasonlít a programnyelvek változóihoz. Itt az

```
x = y
y = z
a := $(x)
```

eredményeképpen az 'a' értéke 'z' lesz, mert 'x'-et kiértékelve 'y'-t kapunk, míg ennek értéke 'z'. (Bizonyos esetekben ez jó, mert nagy szabadságunk van, más esetekben viszont egyszerű dolgokat nem vagyunk képesek megoldani a sima egyenlőségjellel.) Igen hasznos lehetőség a helyettesített hivatkozás, ekkor az

```
obj := x.o y.o z.o
forras := $(obj:.o=.c)
```

végrehajtása után a 'forras' értéke 'x.c y.c z.c' lesz.

Természetesen ezeket a lehetőségeket lehet vegyíteni, és ekkor az

```
a_obj := a.o b.o c.o
1_obj := 1.o 2.o 3.o
source := $(a1)_obj.o=.c)
```

eredményeképpen a 'source' értéke 'a.c b.c c.c' vagy '1.c 2.c 3.c' lesz az 'a1' értékétől függően.

Változóinkhoz további tagokat fűzhetünk a '+=' használatával, például 'obj += w.o', és még vagy húsz függvénnyel variálhatjuk őket, amelyek között olyanok is vannak, mint 'addprefix', 'addsufffix' vagy 'wildcard'. Ezeket a változókat nem csupán fájlnevek helyett használhatjuk, hanem programnevek, opciók, paraméterek helyett is. Húsz, előre definiált programnevet tartalmazó változót ismer a program, így például a CC rejti a C fordító nevét, ezért a C programokat fordító parancsok általában így kezdődnek: '\$(CC)', és ha véletlenül nem a gcc-t használjuk, hanem egy másik fordítót, akkor ezt is elég lesz csak egy helyen megemlíteni.

Vannak olyan esetek, amikor nem elegendő csupán változókat átírni, ekkor a szabálynak a parancs részében feltételes utasításokat tehetünk: 'if***-else-endif', ahol az 'ifeq', 'ifneq', 'ifdef', 'ifndef' feltételeket használhatjuk.

Implicit szabályok

A változók használata hordozhatóvá teszi a Makefile-t, viszont meg is növeli a méretét. (Valamikor azért kezdtük el alkalmazni a rövidítéseket, változókat, hogy ne kelljen sokat gépelni!) Szerencsére kitalálták az implicit, azaz magától értetődő szabályokat is. Elvégre, ha egy object fájl kell készíteni a neki megfelelő C forrásfájlból, akkor egyértelmű, hogy milyen programot kell használni és hogyan, így a szabály leírásából kihagyható a parancs, sőt még a C forrásfájl neve is. Ezzel a korábbi példa a

```
mode.o : mod.h
```

sorra egyszerűsödik. Ez az egyszerűsítés igen kedvelt, a legtöbb Makefile-ban rátalálunk. Van ennél egy még tömörebb írásmód is, viszont mivel ez kevésbé átlátható, ritkán használják:

```
mode.o string.o value.o : mod.h
```

azaz nem a célhoz csoportosítjuk a fájlokat, hanem a célfájlokat a függőkhöz.

Természetesen az implicit szabályoknak is van sorrendjük (amelyet a használt programnyelvek alapján lehet megadni), és így a 'main.o : main.p' hiába szerepel a szabályok között, ha van egy main.c fájlunk is. A sorrend, amelyet nem árt észben tartanunk, ha a kiterjesztéstől eltekintve azonos nevű fájlokról van szó: C, C++, Pascal, Fortran, Ratfor, Modula2, Assembly, Linker, Yacc, Lex, Lint, TeX, Web, TeXin-

fo, Info, RCS, SCCS. (Igen, ezeket mind ismeri a program.)

Ha mi mégis valamilyen más programnyelven kívánunk fejleszteni (például a Clipper famíliából), akkor készíthetünk az implicit szabályhoz hasonló szabályokat:

```
%o : %.prg
      cl $<
```

ahol az '\$<' az első függő fájl, azaz a '%.prg'-nek megfelelőt jelenti (ha jól emlékszem még a Clipperre...). Az előbbihez hasonló, ám parancs nélkül álló szabállyal törölhetjük a neki megfelelő implicit szabályt. A gnumake képes az implicit szabályok láncolására, azaz rájön, hogy ha csak a Yacc fájl van meg, akkor az object fájl elkészítéséhez először egy C fájl kell készítenie.

Rövidíthetünk akkor is, ha hasonló módon hajt végre parancsokat a make:

```
bigoutput : text.g
      generate text.g -big > bigoutput
littleoutput : text.g
      generate text.g -little >
      littleoutput
```

helyett használható a

```
bigoutput littleoutput : text.g
      generate text.g -$(subst
output,,$@) > $@
```

Egyes programnyelveken íródott forrásfájlokat átnézve bizonyos függőségi kapcsolatokat a make önállóan is képes felderíteni, és ezekkel kiegészíteni a Makefile-t.

A make-et elindíthatjuk úgy is, hogy utána megadjunk egy nevet. Ha az egy fájl neve, akkor a program csak azt a fájl próbálja meg elkészíteni. Lehetnek olyan esetek is, amikor ez a cél nem egy fájl neve, például hagyományosan ilyen a 'clean'. Általában van egy olyan szabály, ami ehhez hasonlóan néz ki:

```
clean :
      rm -f *.o
```

tehát törli az összes elkészített object fájl. (Természetesen ebben az esetben semmi értelme a függő fájlok megadásának.)

Hagyományosan ilyen szavak az 'all', azaz készítsen elő minden lényeges fájl; 'clean', azaz töröljön le mindent, amit a make létrehozott; 'mostlyclean', azaz a fontosabb fájlokat azért hagyja meg; 'install', azaz a futtatható fájlokat, a dokumentációkat pakolja a helyükre; 'print', azaz nyomtassa ki a megváltozott forrásfájlokat; 'tar', csomagolja össze a forrásfájlokat; valamint a 'test', próbálja ki a programot. Tekintsük ebből az egyiket:


```
print : *.c
lpr -p $?
touch print
```

Ennek megértéséhez egy kis magyarázat. Szükségünk lesz egy 'print' elnevezésű, esetleg üres fájlra. Ennek csak az a szerepe, hogy tartozik hozzá egy időpont, és amikor ez a szabály végrehajtódik, a 'touch' ezt az időpontot aktualizálja. A '\$?' helyére a megváltozott C fájlok nevei kerülnek, s azokat nyomtatjuk ki. A példában szereplő csillag (mint dzsóker) használata kényelmes, de nem árt vele vigyázni, mert csak a már létező fájlokat jelenti, így könnyen meglepetést okozhat az 'obj = *.o' definíció, hiszen amíg nem készültek el az object fájlok, addig ez egy üres listát ad eredményül. Ezért ilyen esetekben a forrásfájlokra kell használni a dzsóker karaktert (mert azok biztosan léteznek), és abból kiindulva a helyettesítéses hivatkozást az object fájlokra.

Több alkönyvtárral

Természetesen nem kell minden fájl-nak egy alkönyvtárban szerepelnie. Ha több alkönyvtárat használunk, akkor a 'VPATH' beállítható, hogy a program milyen sorrendben kutasson a fájlok után, míg a 'vpath' a kiterjesztés (azaz típus) alapján keresi más és más alkönyvtárakban a fájlokat, például a

```
vpath %.h header
vpath %.c cfile1 cfile2
```

hatására a C forrásprogramokat (ha nincsenek az aktuális alkönyvtárban), először a 'cfile1' — és ha még ott sincs, akkor — majd a 'cfile2' alkönyvtárban keresi. A 'header' alkönyvtárba feleslegesen be sem néz, oda csak akkor kukkant be, ha az aktuális alkönyvtárban nem találja a keresett header fájlt.

A Makefile-ba további fájlokat fűzhetünk be az 'include' utasítással, vagy ha jobban tetszik, akkor újabb make parancsokat is elindíthatunk. Mivel a Makefile-t nemcsak a készítője olvassa el, hanem az is, akinek elsőre nem sikerült elkészítenie a kívánt fájlt, így nem árt számára elmagyarázni, hogy melyik rejtélyes opció mit is takar. A Makefile ezért tartalmazhat megjegyzéseket is, ezek a '#' jellel kezdődnek, és a sor végéig tartanak. Szokás a több hasonló cél megadása helyett kommentezni bizonyos opciókat, amelyek igény szerint a '#' jel törlésével állíthatók vissza.

Noha megpróbáltam csak az érdeke-
sebb vonásokat bemutatni, így is
hosszúra sikeredett ez a leírás — bár
ez is eltörlődik a program mellé adott

több mint százoldalas, és helyenként még így is szűkszavú dokumentáció mellett. Egy-két Makefile elolvasása és megértése után már mi is hozzáfogh-tunk saját Makefile-jaink írásához. Je-lentős időt spórolhatunk meg velük, és több idő jut az érdemi programozásra. Hasznunk származhat abból is, ha csak megértjük az épp „leeftépézet” prog-ram Makefile-ját, és a rendszergazda vagy valamely guru zaklatása helyett mi adjuk ki a 'make documentation' parancsot; és a TeX mindennemű isme-rete hiányában is csak a nyomtatóból kell kivennünk a leírást.

A make segíthet abban is, hogy az otthoni és a munkahelyi gépek között hordozgatott fájlokból mindig az aktu-ális szerepeljen a lemezünkön, és ren-geteg további automatizálható feladat megoldására használhatjuk fel.

A program DOS-os verziója sajnos nem tökéletes másolata az eredetinek, de még így is legalább 286-os gép kell a használatához. A program átültetője próbálja magyarázni a bizonyítványát, 31 pontban fejti ki elmarasztaló véle-

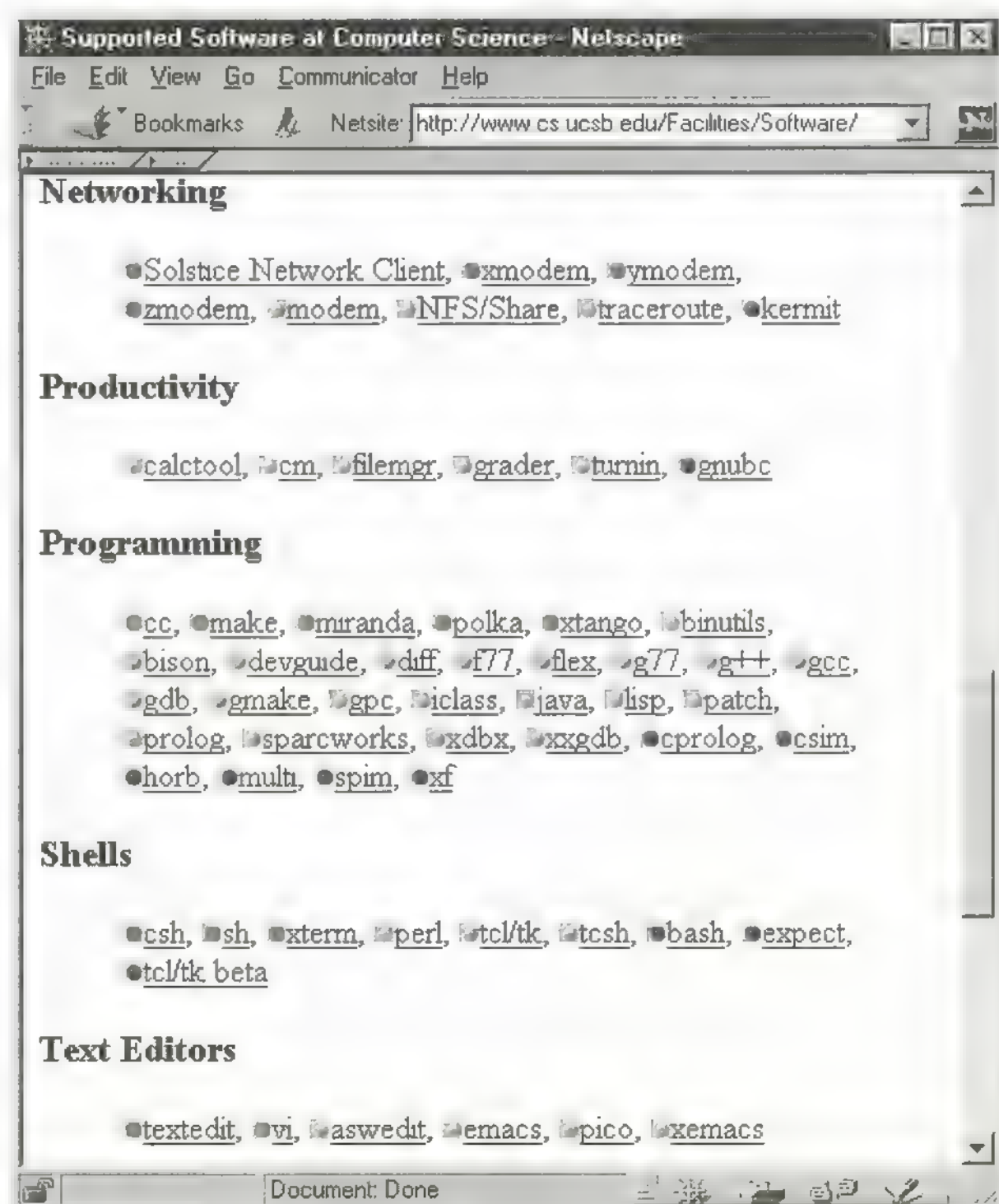
ményét a DOS-ról. Érdemes elolvasni. A lemezmellékletre felkerült anyag épp a minimálisan használható része a cso-magnak, de az ismerkedéshez, elindu-láshoz talán ez is elég lesz.

Amint a CD-n mellékelt fájlokból is látjuk, a Makefile aktuális igényekhez igazítását apró sed programok végzik.

Nem szóltunk még a kukac (vagy bejgli) karakterek (@) közé zárt szavak jelentéséről. Ezek ebben a formában állnak az összes forrásfájlból és doku-mentációban, majd a valódi fordítás előtt cserélődnek le normális alakra. Ez azért van így, mert például egy helyen (egy sed programban) elég lecserélni egy program valamelyik opcióját, és ezzel nemcsak a program használja már ebben az új formában az opciót, hanem rögtön az összes vonatkozó dokumen-tációban is az aktuális információ fog szerepelni.

A mellékelt — igaz, a programnál kicsit frissebb — make.info remélhető-leg majdnem minden fennmaradt kér-désre választ ad.

Aszalós László



Excel — felsőfokon III.

Konstansok, változók, típusok

Az előzőekben láttuk, hogyan lehet létrehozni a makrógenerátorral VBA nyelvű rutinokat, és azt is láttuk, hogy ezek általában némi kiigazításra szorulnak. Ámbár a makrógenerátor ezután is hasznos segítőtársunk lesz, a következő részekben áttekintjük a VBA programok legfontosabb szerkesztési, belövési és futtatási szabályait. Így már önálló rutinokat is meg tudunk majd írni, s ezekből komolyabb alkalmazásokat is fel tudunk építeni.

Tudjuk, hogy VBA szóhasználat szerint az *alkalmazás* egy feladat megoldására szolgáló program, amelyet kényelmi okokból rendszerint egy (példáinkban UAXLMIPn.XLS nevű) Excel munkafüzetben építünk fel, bár nincs akadálya, hogy több munkafüzetre is kiterjesszük. A program *modul(ok)*ból tevődik össze. Egy modul egy modullapon elhelyezkedő deklarációk és *rutinok* — eljárások, illetve függvények — alkotnak, ezeket a továbbiakban közös néven *programelemeknek* nevezzük. A modul logikai szerepe abban nyilvánul meg, hogy a deklarációk érvényességi körét, illetve a rutinok használatát az aktuális modullapra lehet korlátozni, amelyen elhelyezkednek. Természetesen van lehetőség globális szintű — VBA szóhasználat szerint *nyilvános* (**Public**) — deklarációkat, illetve rutinokat is létrehozni.

Mind a modulok, mind a rutinok *fejrészből* és *törzsből* állnak; ez alkalommal a fejrészről lesz szó. (Megjegyzéseket a mondottak szerint bárhol el lehet helyezni, ezt a továbbiakban nem említjük.)

A fejrész *deklarációkat* tartalmazhat. Jóllehet kialakítása nem mindig kötelező, érdemes rászánni egy kis többletmunkát: jelentősen meg tudja könnyíteni a belövést, és emellett az áttekinthetőséget is javítja. A konstansok és a változók három hierarchiaszintet alkotnak: a globálisak érvényességi köre az egész programra (alkalmazásra), a modulszintűeké az aktuális (a deklarációt tartalmazó) modullapra, a lokálisaké egy rutinra terjed ki. Deklarálni az alapértelmezéstől eltérő beállításokat, valamint a globális és a modulszintű konstansokat és változókat kell, a lokális konstansokat és változókat lehet.

Itt jegyezzük meg, hogy egy modul vagy rutin aktiválásakor a VBA a típusnak megfelelő kezdőértékkel inicializálja a változókat, mégpedig első aktiváláskor mindet, újabb aktiváláskor csak a lokálisakat — a sztatikusak kivételével, lásd alább. Mivel a modullapokon a VBA sem tesz különbséget, a következőkben a saját azonosítókat (konkrét konstans-változóneveket) is ezzel a kiemelt betűtípussal jelöljük.

Globális és modulszintű deklarációk

A modulok alapértelmezésben *globálisak*, ami azt jelenti, hogy a globálisnak deklarált konstansok és változók, valamint a rutinok — ezek *alapértelmezésben globálisak* — programozási szinten elérhetők a munkafüzet minden modullapjáról, továbbá minden más állományból is, amelyek szerepelnek az *Eszközők - Hivatkozások* menüpont „Létező hivatkozások” ablakában. Alapvető célkitűzésünknek megfelelően — nevezetesen, hogy megkönnyítsük, illetve rugalmasabbá tegyük az Excel használatát — rutinjainkat mindig Excel-lapokról

fogjuk majd aktiválni, és ennek módzataival a későbbiekben bőven foglalkozunk.

A *modulszintű* deklarációk az aktuális modullapon globálisak, vagyis elérhetők minden rutinból, amelyik az adott modullapon van. A globális és a modulszintű deklarációkat a modullapok tetején, az első rutin előtt kell *leírni*.

a) A modullap alapértelmezéseinek beállítása, illetve megváltoztatása

Option Base 1

Alapértelmezésben a tömbök kezdőindexe 0, a deklaráció hatására 1 lesz; más értékeket nem lehet megadni. Mivel az Excelben a tartományok indexelése 1-gyel indul, célszerű a tömbök indexelést a VBA nyelvben is 1-ről indítani. A továbbiakban ezt az értelmezést tekintjük érvényesnek!

Option Compare {Binary | Text }

Füzérek (karakterláncok), illetve füzéreket tartalmazó változók összehasonlításakor az alapértelmezés szerinti bináris módban a kis- és nagybetűk különbözőek, szöveges módban nem.

Option Explicit

Hatására megszűnik az implicit deklaráció: kötelezővé válik a konstansokat és a változókat előre deklarálni. Az elírásokból, valamint a többszörös névhasználatból adódó hibák könnyebb kiszűrése végett ajánlatos ezzel a deklarációs móddal dolgozni; példáinkban mindig ezt használjuk. Ha az *Eszközők - Egyebek* menüpont választása után a "Basic általános" lapon beállítjuk a "Változó deklarálása szükséges" jelölőnégyzetet, a VBA automatikusan generálja ezt a deklarációt minden újonnan létrehozott modullapon; a már létezők beállítása nem változik.

Option Private Module

A deklaráció hatására a modul zárttá válik más munkafüzetek számára; vagyis globális deklarációit és rutinjait csak az aktuális munkafüzetből lehet elérni.

{DefInt| DefLng| DefSng| DefDbl| DefCur| DefStr| DefVar} *betű* [- *betű*]

Változók alapértelmezés szerinti típusának beállítása. Ha egy *típus nélküli* (l. alább) változó neve a megadott betű(tartományba eső betű)vel kezdődik, akkor típusa az utasításnak megfelelő (rendre: **Integer**, **Long**, **Single**, **Double**, **Currency**, **String**, illetve **Variant**) egyszerű típus lesz (l. 5. táblázat). Típusbeállítás hiányában az alapértelmezés **Variant**, a továbbiakban ezt tekintjük érvényesnek.

A betű(tartomány) előírásánál nincs különbség kis- és nagybetűk között. Ha A—Z tartományt adunk meg, újabb típusbeállításra már nincs lehetőség, és ez az ékezetes betűkkel kezdődő nevekre is vonatkozik. (Konkrét változó

természetesen ebben az esetben is bármilyen más, megadott típusú lehet.)

b) Konstansok deklarálása

[Public | Private] Const konstansnév_
[As típusnév] = kifejezés

A globális (**Public**) konstansok a mondottak szerint érhetők el más modullapokról; **Private** kulcsszóval korlátozhatjuk egyes konstansok használatát az aktuális modullapra (modulszintű deklaráció). Ámbár ez az alapértelmezés, és így ez a kulcsszó elhagyható, a könnyebb áttekintés végett javasoljuk használni.

A konstans típusát ugyan nem kell megadni — a VBA automatikusan illeszti az adattartalomhoz —, a jobb olvashatóság vagy nagyobb pontosság elérése indokolhatja.

Egy sorban — listaelvásztóval — több konstans is deklarálható. A kifejezésekben előzőleg deklarált konstansok is használhatók; körkörös hivatkozás természetesen nem megengedett.

Fűzérkonstanst (pl. állomány, munkalap stb. konkrét nevét) mindig idézőjelek (" ") között kell megadni. Emlékeztetünk még, hogy a beépített Excel-konstansok neve "xl", a Visual Basic-konstansoké "vb" karakterekkel kezdődik.

c) Változók deklarálása

[Public | Private | Dim] névlista_
[As típusnév]

A névlista egy vagy több változónevet tartalmazhat; típusnév nélküli változó(k) az alapértelmezés szerinti típust veszik fel.

A globális (**Public**) változók a mondottak szerint érhetők el más modullapokról.

A **Private** vagy **Dim** kulcsszóval deklarált *modulszinten* *lokális* változók csak az aktuális modullapon érhetők el. A két kulcsszó hatása ekvivalens; a globális-lokális kontraszt kiemelése érdekében az előbbi használata javasolt.

A globális és a modulszintű konstansok és változók mindaddig megőrzik értéküket, amíg a modullap nyitott állományban van.

Egy modullapon a konstansok, változók és rutinok nevének különbözőnek kell lennie. Az esetleges névütközések elkerülése érdekében más modullapon deklarált programelemekre

[állománynév]modullap-név!
programelem-név

szintaxis szerint célszerű hivatkozni. Ha szükség van rá, mert speciális karaktert tartalmaz, az állománynevet szögletes zárójelek közé tesszük!

d) Saját típusok deklarálása

Type saj_típusnév
saját_váltnév1 As típusnév
saját_váltnév2 As típusnév
...

End Type

Típusdeklarációban mind beépített, mind korábban deklarált saját típusok használhatók, akár egyszerű típus, akár fix vagy dinamikus tömb formában. Típusdeklarációk tetszőleges mélységben egymásba ágyazhatók, de a típus teljes mérete nem lehet több 65 535 bájtjánál.

A típusdeklaráció mindig globális; az adott típussal természetesen bármilyen szintű változót vagy konstanst deklarálhatunk. Valamely saját típusú változó esetén az első típuselemre a fenti példa szerint

változónév . saját_váltnév1

szimbólummal hivatkozhatunk.

Megjegyezzük, hogy ez a típusfogalom azonos a C nyelv *struktúra* (struct), illetve a Pascal *rekord* (Record) fogalmával.

Lokális (rutinszintű) deklarációk

A rutinok belsejében *nincs deklarálási kényszer*: a VBA alapértelmezésben *implicit* deklarációkkal dolgozik; magyarárn a számára ismeretlen karaktorsorozatokat változónévként kezeli. Mindenesetre a hibalehetőségek és a helyfoglalás csökkentése, valamint a futási sebesség növelése érdekében célszerű a változókat deklarálni, amit példáinkban mindig meg is teszünk.

Lokális konstansoknak és változóknak lehet a globálisakéval azonos nevük, és ilyen esetekben mindig a legalacsonyabb szintű név van érvényben. (Ha például megváltoztatjuk egy rutinban az ott deklarált lokális *temp* változó értékét, ez nem érinti a modulszintű *temp* változót, mivel ezt az utóbbit ebből a rutinból el sem érhetjük.) Rutinon belül nem lehet globális konstanst vagy változót deklarálni; az *implicit* változók is mindig lokálisak.

a) Konstansok deklarálása

Const konstansnév = kifejezés

Noha a konstansok jellegükben fogva általában globálisak vagy modulszintűek, természetesen lehet lokális konstanst is deklarálni, a megadott szintaxis szerint. (Ahogy említettük, ez a rutin aktiválásakor, az inicializálás során kap értéket.)

b) Változók deklarálása

[Dim | Static] névlista [As típusnév]

A névlista egy vagy több változónevet tartalmazhat; ha nem adunk meg típusnevet, a változó(k) az alapértelmezés szerinti típusúak lesznek.

A **Static** kulcsszóval deklarált változók értéke mindaddig megmarad, amíg a modullap nyitott állományban van, a **Dim** kulcsszóval deklaráltaké a rutinból kilépéskor általában elvész. Ezt elkerülhetjük, ha magát az eljárást deklaráljuk sztatikusnak a

Static Sub eljárásnév([paraméterlista])
szintaxis szerint, ekkor ugyanis a kódja nem a verembe kerül, és így lokális változóinak értéke kilépéskor nem sérül.

Beépített típusok

a) Egyszerű típusok

A konstansok és a változók típusa végső soron mindig a beépített egyszerű típusokra vezethető vissza, ezek legfontosabb jellemzőit az 5. táblázatban foglaltuk össze. A típust az **As típusnév** szimbólum helyett a változónévhez (szóköz nélkül) illesztett rövid jellel is deklarálhatjuk.

Boolean

Látható, hogy nincs valódi (1 bájtos) logikai típus. Ha logikai típusú változót számmá alakítunk, a **True** logikai értékből -1, a **False**-ből 0 lesz; fordítva minden nullától különböző érték **True**-vá alakul. Összehasonlításokban először a logikai változó értékelődik ki; ezért például a

3 = True kifejezés értéke **False**, mivel **3 ≠ -1**.

Integer, Long, Single, Double

Ez a négy numerikus típus a szokásos 2, illetve 4 bájtos egész, valamint egyszeres, illetve kétszeres pontosságú lebegőpontos számok tárolására szolgál.

Currency

Ez a numerikus típus speciális fixpontos vegyes számokat tárol, 11 egész és 4 tizedesjeggyel; valutaértékek gyors és pontos számítására szolgál. A két lebegőpontos típus érték-tartománya ugyan jóval nagyobb, de a számítások során fellépő kerekítési pontatlanságok miatt kevésbé alkalmasak erre a célra, és végrehajtási idejük is jóval hosszabb.

Date

Ez a típus dátumokat és időpontokat tárol **Double** számként: a szám egészrésze a napok számát, törtrésze a

5. táblázat: Egyszerű típusok

5. táblázat: Egyszerű típusok							
Típus		Rövid jel	Hossz (bájt)	Alsó	Felső	Legkisebb (≠ 0) abszolút érték	Kezdő érték
Kód	Név			értékhatár			
11	Boolean	-	2	True (-1)	False (0)	-	False
2	Integer	%	2	-32 768	32 767	1	0
3	Long	&	4	-2 147 483 648	2 147 483 647	1	0
4	Single	!	4	± 3,402823 E38		1,401298 E-45	0
5	Double	#	8	± 1,797693134862315 E308		4,94066 E-324	0
6	Currency	@	8	± 9 223 372 033 685 477,5808		0,0001	0
7	Date	-	8	100/01/01 0:00:00	9999/12/31 23:59:59	-	100/1/1
8	String	\$	1 B/kar	0 karakter	≈ 65500 karakter	-	" "
8	String*n	-	n	-	-	-	" "
9	Object	-	4	-	-	-	Nothing
-	Variant	-	[16]	Értékadás után az aktuális típustól függő			Empty

napon belüli időpontot jelenti; $0 \Rightarrow 1998.$ december 30, 0:00 óra.

A VBA két # között megadott szöveges adatot dátumként, illetve időpontként kísérel meg értelmezni; érvényes formák például:

#98 - 3 - 15#, #12,22,32#, #1848. márc. 15. 12:00# stb.

Excel munkalapokról dátum- és időadatokat biztonságosan csak **Date** típusú változókba lehet átvinni, illetve ilyenekből lehet az Excel által is felismerhető dátumokat visszaírni. Néhány gyakoribb dátum- és időfüggvényt a 6. táblázatban adunk meg.

String, String*n

Változó (legfeljebb $\approx 65\,535$), illetve n karakter hosszúságú füzér (karakterlánc); a fix hosszúságúak szükség esetén jobbról szóközökkel feltöltve.

Az **LTrim(szövegkifejezés)**, **RTrim(szövegkifejezés)**, illetve **Trim(szövegkifejezés)** függvény segítségével eltávolíthatjuk a felesleges szóközöket a szöveg elejéről, végéről, illetve mindkét oldaláról.

Object

Ha egy ilyen (vagy **Variant**) típusú változót **Set** paranccsal hozzárendelünk egy objektumhoz, ettől kezdve tartalmazza az objektum címét, és minden tekintetben helyettesíti azt.

Mivel rendelkezik az objektum metódusaival és saját-ságaival is, az objektumot a változón keresztül manipulálhatjuk.

Itt jegyezzük meg, hogy diszjunkt tartományokat a **Union** metódussal építhetünk össze. Egy mintapélda:

```
Dim tart, halmaz As Object, _
    darab As Integer
set tart = ActiveSheet.Range("A1:D5")
set halmaz = Union(tart, ActiveSheet_
    .Range("C4:F9"))
halmaz.Select
darab = Selection.Areas.Count
Set tart = tart.Resize(4, 2)
```

Az egyesített tartományt a halmaz változón keresztül jelöljük ki; a kijelölést az **Areas** metódussal összetevőire bontjuk, amelyek darabszámát a **Count** sajátosság adja. Így a programrészlet végrehajtása után a darab változó értéke 2 lesz.

Az utolsó utasítással megváltoztatjuk az A1:D5 tartományt A1:B4 re (4 sor, 2 oszlop). A hozzárendelést az

objektumváltozó = Nothing

utasítással szüntethetjük meg.

Vegyük észre, hogy a fenti példa második utasításával szemben a

```
tart = ActiveSheet.Range("A1") [.Value]
utasítás (.Value elhagyható) csak az A1 cella tartalmát
veszi át a tart változóba, objektumtulajdonságait nem.
```

Variant

A VBA nyelv legravaszabb típusa: bármelyik másikat helyettesítheti, tartalmát átveheti, illetve felülírhatja — sőt

6. táblázat: Dátum- és időfüggvények

Függvény	Eredmény
Now	Aktuális dátum és időpont Double számformában
Date, Date\$	Rendszerdátum dátumszám, illetve 10 karakteres szöveges formában
Dateserial(év, hó, nap)	A megadott számhármassá alakítása dátumszámmá
Datevalue(szövegkifejezés)	Dátumként értelmezhető szöveg átalakítása dátumszámmá
Time, Time\$	Rendszeridő időszám, illetve 8 karakteres szöveges formában
Timeserial(óra, perc, másodperc)	A megadott számhármassá alakítása időszámmá (Double törtrésze)
Timevalue(szövegkifejezés)	Időpontként értelmezhető szöveg átalakítása időszámmá

magában foglalhat tetszőleges, akár **Variant** típusú és tetszőleges méretű tömböket is. Értékadáskor a megfelelő altípusba megy át; ezeket **VarType** *n* szimbólummal is szokás jelölni. (Egyszerű típusok esetében *n* az 5. táblázatban szereplő típuskód; idegen adattípus esetében 13. Az altípusoknak a műveletvégzés során van jelentőségük.)

Fontos: Saját típusú változók vagy ilyeneket tartalmazó tömbök nem lehetnek **Variant** típusú változó elemei!

Ha nem állítunk be mást, az implicit és a típus nélkül deklarált változók alapértelmezésben **Variant** típusúak; tartalmuk az első értéadás előtt **Empty**; ebben az esetben altípusuk **VarType** 0. (Megjegyezzük, hogy létezik még egy speciális üres altípus is, a **VarType** 1, ennek értéke **Null**. Mivel jelentősége csak egyes adatbáziskezelő rendszerekben van, a továbbiakban nem használjuk.)

A **Variant** típus speciális alosa az **Error** altípus (**VarType** 10); ennek használatával a hibakezeléssel kapcsolatban foglalkozunk majd.

A típus hossza legalább 16 bájt, amely más típusú adatok felvételekor automatikusan illeszkedik az adattípus hosszához; füzerek esetében például karakterenként 1 bájjal nő. További előnye, hogy **Variant** típusú változóval végzett művelet esetén a VBA automatikusan elvégzi a szükséges típuskonverziókat is. Nézzük a következő példákat:

```
Dim teszt
' A változó alapértelmezés szerint Variant típusú
teszt = "100"
' Tartalma egy 3 karakteres füzér ("100")
teszt = teszt-99
' Tartalma egy 2 karakteres egész szám (1)
teszt = teszt & "ház"
' Tartalma egy 4 karakteres füzér ("1ház")
teszt = teszt * 3
' Hiba: szöveggel nem végezhető aritmetikai művelet!
```

A típus alkalmazásának hátránya, hogy gyakran — mint a példákban is — feleslegesen sok helyet foglal, és lényegesen lassítja a műveletvégzést; ezért megfontoltan használjuk!

Futás közbeni hibák elkerülése végett célszerű lekérdezni a **Variant** változó aktuális altípusát; erre a következő függvények állnak rendelkezésre:

```
TypeName (változónév)
Visszaadja az aktuális típus nevét;
isArray(változónév)
True értéket ad, ha a változó tömböt tartalmaz
IsDate (változónév)
True értéket ad, ha a változó Date altípusú
IsEmpty (változónév)
True értéket ad, ha a változó még nem kapott értéket
IsError (változónév)
True értéket ad, ha a változó Error altípusú
IsNumeric (változónév)
True értéket ad, ha a változó numerikus altípusú
```

b) Tömbök

Tömböket a változónév után kerek zárójelek közé írt indexhatár(ok) és az elemtípus megadásával lehet deklarálni, a változók deklarálási szabályai szerint. Index csak numerikus típusú lehet; a dimenziószám legfeljebb 60.

Indexhatárok megadása nélkül *dinamikus tömbök* hozhatók létre; ezek indexhatárait csak rutinon belül (futás során) kell definiálni feltöltésük előtt, **Redim** kulcsszóval. Az egyszer megadott dimenziószám nem változtatható, de az indexhatárok tetszőlegesen újradeklarálhatók; ilyenkor a tömb aktuális tartalma általában elvesz. Ha meg akarjuk tartani a tömbértékeket, csak az utolsó dimenzió felső határa változhat.

Variant típusú változókhoz az **Array** függvénnyel rendelhetők tetszőleges típusú és méretű tömbök. (A változó altípusának kódja **Variant** típusú tömbök esetében 12, egyéb tömbök esetén 8192 lesz.)

Excel tömböket, azaz cellatartományokat csak **Variant** típusú tömbökhöz rendelhetünk hozzá, mégpedig

— egy sorból, illetve egy oszlopból álló Excel tömböt egydimenziós,

— téglalap alakú tartományt kétdimenziós tömbbé alakíthatunk. Fordított irányban nincs típuskorlátozás: bármilyen típusú egydimenziós VBA tömbbel egy cellasort, kétdimenzióssal téglalap alakú Excel tartományt tölthetünk fel.

Néhány példa tömbök deklarációjára:

```
Dim összeg(10) As Long
```

```
Dim összeg(0 To 9) As Long
```

```
Dim összeg(100 To 109) As Long
```

Egydimenziós, 10 Long típusú elemből álló tömbök; az indexhatárok rendre 1..10, 0..9, illetve 100..109.

```
Dim tömb(7)
```

7 Variant típusú elemből álló egydimenziós tömb, indextartománya 1..7.

```
Dim napok
```

```
napok = Array("Hétfő", "Kedd", "Szerda", "Csütörtök", "Péntek", "Szombat", "Vasárnap")
```

Variant típusú változó, 7 elemű, String típusú tömbbel feltöltve; ettől indextartománya 1..7 lesz.

```
Dim tmb(4, 11 To 22) As Boolean
```

Kétdimenziós, 48 elemű, logikai típusú tömb; indextartományai 1..4, illetve 11..22.

```
Static számok() As Integer
```

```
Redim számok(1 To 3, 2 To 6, 4)
```

```
Redim számok(x,y,z)
```

```
Redim Preserve számok(x,y,u)
```

Dinamikus egész típusú tömb, amelyet először 3 dimenziós 60 eleművé, majd az x,y,z, illetve az u változó aktuális értékének megfelelő elemszámúvá deklaráltunk át; az utolsó esetben az előző tartalmat is megőrizve. (Lásd a fenti megjegyzést!)

A deklarációk használatára a következő folytatásokban mutatunk példákat.

Álló Géza

VÍRUSÓRJÁRAT rovatunk
anyagtorlódás miatt most kimaradt,
de szeptemberi számunkban a vírusok
elleni védekezés lesz A HÓNAP TÉMÁJA,
amelyhez a CD-n is gazdag antivírus
összeállítást mellékelünk.

Egy programozási nyelv változásai

Redundancia nélkül

A Fortran 90 nyelvet úgy definiálták, hogy a felbecsülhetetlen értéket képviselő régi Fortran programokat gyakorlatilag változtatás nélkül fel lehessen dolgozni az új fordítóprogramokkal. Ez természetesen lehetőséget kínál egy-egy programlépés többféle formában történő leírására. Új programok fejlesztéséhez egy forma is elég. A második, harmadik stb. redundáns.

Az Új Alaplap 13. (1995-ös) évfolyamában megjelent cikksorozatból az olvasó megismerkedhetett a Fortran 90 nyelv szintaxisával, szemantikájával. A programozó stílusától függ, hogy egy-egy konkrét esetben a szemantikailag egyenértékű, eltérő szintaktikájú változatok közül melyiket választja. Talán a legegyszerűbb példa egy változó kezdeti értékének deklarálása. Az, aki szószervezésű gépeken kezdett programozni, és az akkor divatban volt kezdőbetű szerinti automatikus típusdeklaráció híve, ezt írja:

```
DOUBLE PRECISION D
DATA A,D,I/1.23,4.56D7,89/
```

Aki fél a gépelési hibáktól, ezért minden változót deklarál (a deklarátlanokat felismeri a fordítóprogram), valami ilyesmit programoz:

```
IMPLICIT NONE
REAL A
DOUBLE PRECISION D
INTEGER I
DATA A,D/1.23,4.56D7,89/
```

A fiatalabbak, akiknek nincs emléküik az IBM System/360 előtti időkről, kissé másképp fogalmazzak:

```
IMPLICIT NONE
REAL*4 A
REAL*8 D
INTEGER*4 I
DATA A,D/1.23,4.56D7,89/
```

Végül azok, akik az Új Alaplap cikksorozatát is tanulmányozták, ilyen programot írhatnak:

```
IMPLICIT NONE
REAL (KIND=4) :: A
REAL (KIND=8) :: D
INTEGER (KIND=4) :: I
DATA A,D/1.23,4.56D7,89/
```

vagy akár ilyent:

```
IMPLICIT NONE
REAL (KIND=4) :: A=1.23
```

```
REAL (KIND=8) :: D=4.56D7
```

```
INTEGER (KIND=4) :: I=89
```

A fenti formaváltozatok rendre megfelelnek a Fortran II (kb. 1960), a Fortran 66 (Fortran IV), Fortran 77 és a Fortran 90 nyelvi változatok közzététele idején korszerűnek tekintett programozási stílusoknak. A Fortran 90 mindegyiket elfogadja. Ezt a sokrétűséget a nyelvi szabványt kidolgozó munkabizottság azzal indokolta, hogy ily módon elkerülhető a több évtizede kidolgozott, esetleg 100-200 ezer forrásnyelvi sorból álló, óriási értékű programok átkódolási munkája (és a közben ejtett hibák felderítésének gyötrelme).

Nyilvánvaló, hogy egy olyan fordítóprogram, amely ugyanarra a programrészletre vonatkozóan ilyen sokféle leírási formát elfogad, nagy és bonyolult, tehát szükségképpen lassú. Új programok írása esetén nincs is szükség arra, hogy a változatos stílus érdekében „válogatni” lehessen a kódolási lehetőségek között. Ezt meg kell hagyni a szépirodalom művelőinek.

Nem meglepő, hogy a Lahey Fortran 90 fordítóprogramjának 1.10 változata kereken 9 MB helyet foglal el a merevlemezen (az újabbak, amelyek támogatják például a windowsos programok írását is, természetesen sokkal többet). A programírási változatokat meg nem engedve viszont másfél MB-tal kevesebb elegendő! Ez a fordítóprogram-változat az ELF 90 (Essential Lahey Fortran).

Az ELF 90 legfontosabb tulajdonságainak alábbi összefoglalója illusztrálja ennek a fordítóprogramnak (és ezáltal tulajdonképpen egy programozási nyelvnek) céltudatos fejlesztését.

A forrásprogram olvashatóságát segíti, hogy csak a szabad (azaz nem kártyakép) forma használható, soronként csak egy utasítással. (Folytatósorok természetesen használhatók.) A hibalehetőségeket csökkenti, hogy az IMPLICIT NONE utasítás használata kötelező.

Nagy programoknál gyakori hiba, hogy az alprogramok hívása formális paramétereik típusától eltérő aktuális paraméterekkel történik. Az ilyen hibák elhárítása rendkívül nehéz, hiszen a memóriában található bitkombináció sokféleképpen értelmezhető, tehát könnyen átcsúszhatnak a teszteken. Éppen ezért az INTERFACE utasítás használata kötelező, a paraméterek szerepét pedig az INTENT specifikációval kell megadni.

Hasonlóan nehéz felderíteni azokat a programhibákat, melyek ugyanannak a memóriarekesznek több néven, vagy ellenkezőleg, ugyanazon a néven több memóriarekesznek a felhasználásával függenek össze (ehhez elegendő egyetlen

hibás billentyűleütés!). Éppen ezért tilos az EQUIVALENCE utasítás.

A típusdeklarációkat a bevezető utolsó két példájához hasonlóan a „négyponthos” formában kell megadni. Ez egyebek közt azzal jár, hogy az önállóan is megfogalmazott ALLOCATABLE, DATA, DIMENSION, INTENT, PARAMETER, POINTER, SAVE és TARGET utasítások nem használhatók. A DOUBLE PRECISION szinonima, ezért felesleges. Példa:

```
DOUBLE PRECISION R,Z
```

```
DIMENSION R(5,6,7),Z(:)
```

```
ALLOCATABLE Z
```

```
DATA R/210*8.9/
```

helyett

```
REAL (KIND=8), DIMENSION (5,6,7) :: R=8.9
```

```
REAL (KIND=8), ALLOCATABLE, DIMENSION (:) :: Z
```

írandó.

A Fortran 90 nyelv definíciójában „elavult”-nak (obsolescent) minősített, de megengedett szerkezeteket az ELF 90 tiltja. Nélkülük jobban strukturált programok írhatók. A tiltott utasítások: ASSIGN, CONTINUE, ENTRY, „assigned” GO TO, kiszámított GO TO és alternatív RETURN.

A ciklusutasítások közül az elavult „címkés” DO utasítás helyett a DO – END DO szerkezet alkalmazandó. A kerekítési hibák gépfüggőek; igen „érzékeny” erre a valós (REAL) változós ciklusutasítás: a ciklus más gépen nem ugyanannyiszor futna le. A programok transzportja érdekében tilos a nem egész (INTEGER) ciklusváltozó, illetve ciklusvezérlés. A Fortran 90-nel bevezetett modulokat alkalmazva — szintén a jobb strukturáltság jegyében — feleslegessé váltak a COMMON blokkok, továbbá a kizárólag kényelmi szempontból bevezetett INCLUDE sorok.

A programok transzportját segíti, hogy az eltérő számbrázolású gépeken sincs szükség a szabványos függvények átnevezésére: a fordítóprogram dolga, hogy az argument típusa szerint kiválassza a „közönséges”, „dupla pontosságú”, „komplex” stb. függvényt, melyeket egyformán lehet jelölni. Ilyen esetben az exponenciális függvényt régen például rendre EXP, DEXP és CEXP néven hívták. A Fortran 90-ben (és így az ELF 90-ben is) egyformán EXP-ként jelölik mindegyiket, ami egyébként a megszokott matematikai jelöléshez is közelebb van.

Az utasításfüggvények használata eredetileg is korlátozott volt. Helyettük (és velük együtt) a programszegmenst logikailag lezáró STOP vagy RETURN utasítást követően, a CONTAINS alapszó után lehet függvényeket, alprogramokat deklarálni, melyek lokálisak az őket tartalmazó szegmensben.

A PAUSE utasításnak PC-ken nincs létjogosultsága, hiszen az operációs rendszerrel közölni kívánt operátori parancsot az Enter billentyű leütése zárna, ami viszont a PAUSE-t érvényteleníti. A PC-kre kifejlesztett ELF 90 nem ismeri a PAUSE-t.

A fenti és (a felsoroltakhoz képest lényegtelen) egyéb megszorításokat betartva írt ELF 90 programok teljesen szabályos Fortran 90 programok, tehát bármilyen gépre, bárki által írt fordítóprogrammal lefordíthatók. Ez azt is jelenti, hogy az ELF 90 igen jó PC-s segédeszköz azoknak, akik bonyolult algoritmusokat kénytelenek kezdőként (?) programozni, például műszaki egyetemi vagy fizikus hallgatók, hogy azután programjukat nagyobb teljesítményű munkaállomáson futtassák. (A Fortran 90 nyelvre vonatkozó részletek megismerhetők az említett cikksorozatból.)

Befejezésül egy kellemes hír: az ELF 90 díjmentesen letölthető a www.lahey.com címről. Összehasonlításképpen: egy „komplett” Fortran 90 fordítóprogram ára megközelíti az ezer dollárt.

Szondi Egon János
szondi@reak.bme.hu

Applixware, Caldera OpenDOS, Caldera OpenLinux, FreeBSD, Linux Developer's Resource, Linux Journal, Motif for Linux/FreeBSD, Official Debian Linux, Red Hat Linux (Alpha/Intel/Sparc), Red Hat's Triteal CDE, SCO UNIX, Slackware Linux, StarOffice, Unifix Linux * * * 40,000-es szakkönyv-adatbázis! On-line rendelés, több ezer könyv-ismertető! Oktatási intézményeknek, diákoknak, könyvtáraknak, viszonteladóknek kedvezményes szakkönyv árak.

SoftWare Station

1111 Budapest, Karinthy F. út 25
Tel/Fax: 371-0704; Tel: 209-5951

Angol nyelvű számítástechnikai szakkönyvek és Linux disztribúciók legnagyobb választéka

<http://www.swsbooks.hu>

Cég	Info#	Old.
Albacomp	01	38.
Allegro	02	04.
Array Data	03	04.
Borland	04	04.
ComputerBooks	05	72.
DIT Digitáltechnika	06	32.
Elender	07	B2.
Hungexpo	08	06.
Infopen	09	71.
Keszo	10	74.
Microsoft	11	42.
Profon	12	72.
Qwerty	13	B3.
Ready	14	72.
Reflex	15	04.
Software Station	16	67.
Telnet	17	B3.
VAR	18	72.
VTCD	19	B4.

**SZOFTVEREK
SOKSZOROSÍTÁSA
FLOPPYRA,
RÖVID
HATÁRIDŐVEL**



Részletes feltételek
az Új Alaplap
szerkesztőségében,
Megyes Zsuzsánál,
telefon: 322-4417.

MIKROBAZÁR

A Mikrobazár rovatban a nem kereskedelmi célú egyéni hirdetések közlése ingyenes.

A kereskedelmi célú apróhirdetések tarifája gépelt soronként (azaz 60 karakterenként) 300 forint.

A terjedelem alapján így kiszámított összeget kérjük átutalni az Új Alaplap Kiadói Kft számlájára (OTP, 11706016-20788599), vagy feladni postai utalványon a kiadó címére (1539 Budapest, Pf. 571), és feltüntetni, hogy „Új Alaplap, apróhirdetés”. A befizetést igazoló szelvény másolatát — a hirdetési szöveggel együtt — a szerkesztőséghez (a kiadóéval azonos címre) küldjék el.

Szerzői jogokat sértő szoftverhirdetéseket nem közlünk le.

Bármilyen típusú szöveg fordítását vállalom angolról magyarra, magyarról angol nyelvre, illetve vállalom kiadványok látványtervezését, szerkesztését is. Cím: Lachner Zoltán, 1195 Budapest XIX., Jahn Ferenc u. 14/a. Telefon: 157-0308.

OBJECTS 2.0 — objektumorientált programozás CLIPPER-ben. Tájékoztató kérhető az alábbi címen: Szűcs János, 4400 Nyíregyháza, Vasvári Pál u. 37. Tel.: (42) 437-331 vagy 465-666/1382-es m.

Adatmentés CD-re, streamerre; winchesterről, floppyról. Ugyanitt beszerzési tanácsadást, hálózattervezést és programkészítést is válllok. Cím: Kovács Lajos, 1031 Budapest III., Vízimolnár u. 10. IV/33.

Alaplapcsere, memória-, winchester- és floppybővítés a helyszínen. MegaSoft. Telefon: 295-5085.

Stúdiómban megbízhatóan, ellenőrzött lefordítom angol, német, francia és magyar nyelvről/nyelvre műszaki és közgazdasági folyóiratok cikkeit, hardver- és szoftverleírásait. Áfás számlát állítok ki.

Cím: Szász György, 1035 Budapest III., Kórház u. 25. Tel.: 168-4874.

Akarod, hogy ingyen tiéd legyen az évszázad viccgyűjteménye? Nos, ha igen, akkor írd az alábbi e-mail címek valamelyikére: kgb@server.gaboraron-misk.sulinet.hu vagy kgb11@freemail.c3.hu vagy qbolusyj@gold.uni-miskolc.hu vagy küldj egy üres kislemet és egy felbélyegzett válaszbortitkot a postacímemre: Kovács Gábor, 3502 Miskolc II., Pf. 83. Telefon: (20)703-438.

Eladó 60 db 1,44-es FD (8e), 1 db Canon BJ-30 nyomtató (40e), 1 db Mitsumi FDD (4e), 1 db Mouse egér (4e), valamint megkímélt állapotban Kossuth és ComputerBooks könyvek. Postacím: Jakab Tamás, 4700 Mátészalka, Nagybányai u. 34. (Fizetés: előre, rózsaszín csekken.)

A Gyermekvilág '94 Alapítvány iskolai múzeuma adományként elfogad, vagy jelképes áron átvesz régi információs eszközöket, számítógépeket, részegységeket, tartozékokat. Az adományokról értékbécselés alapján — kérésre — adócsökkentő igazolást adunk. Cím: 8354 Karmacs, Szent Anna tér 3. Tel.: (83)372-011, este: (83)372-026.

Bármilyen típusú szöveg begépelését, szerkesztését, valamint szórólapok tervezését (nyomtatását 1 példányig) vállalom. Ugyanitt keresek PC-s teletext és bármilyen grafikus képlementő programot. Postacím: Jakab Tamás, 4700 Mátészalka, Nagybányai u. 34.

Óriási kedvezmény! Visual Studio 97 Enterprise Edition komplett új programcsomag (Visual Basic 5.0, Visual C++ 5.0, Visual J++1.1 stb.) féláron eladó. Telefon: 220-4701.

CADKEY '97 Release 2, új, bontatlan csomagolású, jogtiszta szoftver (nem diákverzió) rendkívül jutányos áron, sürgősen eladó. Irányár: 49 000 Ft. Érdeklődni: ifj. Béres Miklós (52)364-845.

Keresek használhatatlan, leselejtezett hardverelemeket (nyomtató, alaplap stb.) lehetőleg ingyen (oktatási célokra). Cím: Kovács Gábor, 3526 Miskolc I., Kassai u. 74. Telefon: (20)703-438.

SZEPTEMBERBEN A HÓNAP TÉMÁJA:

ANTIVÍRUS PROGRAMOK

A Visual Basic házatájáról

Felégetett hidak

A Visual Basic fejlődését mindig megkülönböztetett figyelemmel kísértük. Leginkább azért, mert a Visual Basic mutatta meg az egyik lehetséges kiutat a programozás ijesztő elbonyolódásából. Tájékoztatásunk révén remélhetőleg kevesebben lesznek olyanok, akik mást kapnak, mint amit vártak. Egyaránt csalódhat ugyanis az, aki valami gyerekesen könnyű dolognak véli az önálló programok írását VB-ben, és az is, aki blikkfangos ígéretekkel elkápráztatva minden eszközt készen akar megkapni ebben a programban.

Különösen az 5.0 verzió megjelenése keltette fel sok programozó érdeklődését. Volt, aki addig lenézte a VB-t, vagy azért, mert túlságosan primitív programnyelvnek gondolta, vagy mert tudta róla, hogy hatékonysága még nem megfelelő. Az érdeklődés fokozódása a róla megjelenő művek növekvő számán is lemérhető. Majdnem azonos a témáról egy kiadó ugyanabban az évben ritkán ad ki több könyvet. Most a Visual Basic-kel ez két ízben is megtörtént.

1997. decemberi számunkban foglalkoztunk egy eredeti magyar művel, amely a VB 3.0 programozásába vezeti be az olvasót, valamint egy amerikai szerzőpáros könyvével a Visual Basic 4.0 változatáról (Demeter M. Ibolya: „Visual Basic, Lépésről lépésre”; G. Cornell — T. Strain: „Visual Basic 4, Tippek és trükkök”). Demeter M. Ibolya közben újabb könyvet írt a Visual Basic megismertetésére, egy szűkebb területen, az adatbázisok kezelésében (Demeter M. Ibolya: „Adatbázis-kezelés Visual Basic-ben, Lépésről lépésre”). Könyvének ismertetésére máskor majd visszatérünk). Szinte egy időben jelent meg két másik könyv is a VB 5.0-ról, két olyan szerzőtől, akik néhány éve közösen írtak könyvet az Excel 5 programozásáról. Akkori könyvük címe („Az Excel 5 programozása Visual Basic nyelven”) némi félreértést is okozhatott, hiszen az Excel 5 programozói nyelve nem a szoros értelemben vett a Visual Basic, hanem annak egy sajátos dialektusa, a Visual Basic for Applications (VBA).

A VBA-nak, az Excel programozására kidolgozott programnyelvnek a megjelenése egy nagyobb kísérlet első része. A Microsoft agytrösztje azt igyekezett kipróbálni, hogy életképes lehet-e egy Basicen alapuló egységes programnyelv olyan jellegű feladatok-

ra, mint például programcsomagok együttműködésének jobb megszervezése, rutinalkalmazások automatizálásának elősegítése stb. A kísérlet terepéül az Office programok családi együttesét választották ki. Az Office 97 megjelenésével ez a kísérlet a második szakába lépett. Ebben mutatkozott be a VBA továbbfejlesztett, általánosított változata az egész „Office-család”, az Excel, a Word és a PowerPoint közös nyelveként. Aligha kétséges, hogy az „alkalmazott VB” további kiterjesztésére is vannak tervek, hiszen Bill Gates régi álma a Basic széles körű elterjesztése.

Szerzőpáros — egyéniben

A VB-ről szóló két új könyv szerzőinek, Hargittai Péternek és Kaszanyiczki Lászlónak gazdag dokumentációs anyag állt rendelkezésére, de műveik hiánytalanok vagy ellentmondásmentesnek nem mondhatók, és a leírás sem mindenben felel meg a rendszer működésének. Az is hasznosabb lett volna, ha jobban együttműködve rágják át magukat az anyagon, és például közösen alakítják ki a magyar terminológiát.

Tény az is, hogy eltérő megvilágításban tárgyalják lényegében ugyanazt az anyagot. Hargittai könyve elsősorban a referencia-kézikönyvek közé lenne sorolható, Kaszanyiczkié pedig a felhasználói kézikönyvek stílusában íródott. Így a két mű — a természetes átfedések ellenére — jól megfér egymás mellett, nem teszi egymást feleslegessé. Sajnálatos viszont, hogy ez a különbség a könyvek címéből nem derül ki világosan. Főleg Hargittai könyvének címe kifogásolható, mert akit a programozási nyelv érdekel, azt egyrészt nem elégíti ki a nyelv elemeinek lexikonszerű ismertetése, bármilyen részletes legyen is az; másrészt CD-mellékletével együtt

többet nyújt, mint amit a címe ígér: nemcsak a Learning Edition leszűkített körébe tartozó nyelvi elemek ismertetését közli, hanem kinyomtathatóan közreadja a könyv következő kötetét is, az ActiveX elemek és az AddIn modell hasonló stílusú leírásával.

Egyszerűsödik-e a világ?

Aki fejest akar ugrani a Visual Basicbe, annak tudnia kell, hogy a 4.0 változat még csak a lehetőségét adta meg a 32 bites processzorok kezelésének, az 5.0 viszont már egyenesen elvágja az utat visszafelé, a 16 bitesekhez. Legpregnansabban a karakterláncok kezelésében és a DLL-ek világán látszik meg a tudatos törekvés az előzményekkel való leszámolásra, ugyanakkor azonban az is kiviláglik, hogy még az új koncepció sem tekinthető kiforrottnak. Íme néhány adalék a „felégetett hidak” módszerének megnyilvánulásaira Kaszanyiczki László könyve alapján. (Megjegyzéseiből természetesen senki ne vonja le azt az elhamarkodott következtetést, hogy a szerző kétségbe akarja vonni a rendszer érdemeit.)

1. Más szabályok vonatkoznak a 16 bites Visual Basichez való DLL-ekre, mint a 32 biteshez valóakra. Kaszanyiczki László szavait idézve: „A különbség nem sok, csak annyi, hogy ne lehessen a 32 bites Visual Basichez 16 bites DLL-t használni, és viszont.” Nesze neked, kompatibilitás!

2. A DLL eljárások Visual Basicen belüli deklarációjában is biztosítva van a kettősség, már a Windows szintjén, hiszen a 32 bites változat „nagybetűérzékeny”, megkülönbözteti a kisbetűket a nagybetűktől. Ez a hibaforrás azért tekinthető alattomosnak, mert a Visual Basicre egyébként nem jellemző ez a megkülönböztetés.

3. A 32 bites Visual Basic belsőleg két bájtos kódolásban tárolja a karaktereket, akárcsak a 32 bites Windows a maga Unicode karaktereivel. Egy DLL eljárásnak történő átadás esetén azonban automatikusan egybájtos ANSI karakterekké alakítja őket. („Hogy a dolog még bonyolultabb legyen...” — jegyzi meg Kaszanyiczki rezignáltan.)

4. Már most is sok az összehangolatlanság az API függvények területén a karakterláncok kezelésében. Még a 32 bites Windows változatokhoz készült API függvények is háromféleképpen lehetnek abból a szempontból, hogy milyen típusú karakterláncokat fogadnak el. Van olyan, amelyik csak ANSI típusút, van, amelyik csak Unicode típusút, és van, amelyik mindkét fajtát.

5. A Windows 95 belsőleg egybájtos ANSI ábrázolást használ az API függvények kezelésében, a Windows NT viszont kizárólag Unicode típusút. Mivel a Visual Basic 32 bites változata automatikusan minden karakterláncot ANSI típusúra alakít, külön ügyeskedés kell ahhoz, hogy ezt az automatikus konverziót meg lehessen akadályozni.

6. Az automatikus konverzió elkerülésének legjárhatóbb útja az, hogy speciális Unicode DLL-eket hozunk létre. Ez annyit jelent, hogy az ODL objektumleíró nyelv segítségével típusárát készítünk minden stringkezelő DLL-hez. Jelenleg azonban a Unicode DLL-ek csak a Windows NT alatt képesek működni.

Karakterláncok

Számos probléma adódik abból is, hogy minden változatban újabb típusú karakterláncok tűnnek fel vagy tűnnek el. Egy darabig az új verziók kezelni tudják az előzőt, de egy idő múlva szép lassacskán elfeledkeznek róluk. Néhány példa ezekre:

— A VB 3.0-ban API függvényekkel lehetett módosítani a HLSTR típusú karakterláncokat. A 4.0 változatban ezeket le kellett cserélni az OLE API függvényeire, több függvénynek azonban hiányzott a megfelelője (VBDe-refHlstr, VBDe-refHlstren, VBDe-refTermHlstr).

— A VB 3.0-ban öröndetes módon megjelentek az LPSTR típusú karakterláncok, a C típusú, 00h-val lezárt karakterláncok megfelelőiként. Némi irányváltás következett be a 4.0-ban, amikor bevezették a változó hosszúságú karakterláncok belső manipulálására a BSTR típust, amely karakterláncra mutató pointerként működik (noha a pointer típusú változókat tudatosan kihagyták a Visual Basicből). A karakterlánc hosszát jelenleg a BSTR típus az első karakter előtt tárolja, de ez a megoldás sem tekinthető véglegesnek. A Microsoft már előre figyelmeztet, hogy a jövőben ez változhat.

— Máshol is adódhat probléma a BSTR karakterláncokkal. Jelenleg a hossz tárolása miatt ezek 00h karaktereket is tartalmazhatnak. Amelyik viszont nem tartalmaz, az egyelőre C típusú, 00h bájtal lezárt karakterláncként is kezelhető. Ez a helyzet is megváltozik, mihelyst más lesz a felépítése a BSTR karakterláncoknak.

— A 4.0 verzió előtt hiányzott a Visual Basicből a Byte típus, helyette a String típust kellett használni bináris adatok tárolására. A VB 4.0 16 bites változata továbbra is megengedi ezt, de

a 32 bites már nem. Ez kéretlenül nekiesik a String típusú változóknak, és automatikusan átkonvertálja őket Unicode-ról ANSI karakterekké, akkor is, ha DLL eljárásnak akarjuk átadni, akkor is, ha fájlba írjuk őket.

— Hasonlóképpen abból is katasztrofára származik a VB 32 bites változatban, ha pufferként használjuk a String változókat. Helyette a Byte típust vagy annak tömbjét lehet használni.

Tanulság: őrizzük meg mindig a forráshelyű változatot, bőséges kommentárokkal, és soha ne reménykedjünk abban, hogy hibátlan programjaink az új verzióban is (vagy más rendszerében) változatlan formában futni fognak!

Egy medve három bőre

Sok bosszankodás származhat abból, ha utólag vesszük észre, hogy egy adott programverzióknak nem azt az „változatát” vásároltuk meg, amelyikre szükségünk lenne. A „kezdő”, „középhaladó”, „haladó” rejtjeles kódolásra itt a „Standard”, a „Professional” és az „Enterprise” elnevezéseket használják, de hogy könnyebben eltévedjünk közöttük, a kezdő változatot más alkalommal „Learning Edition” névvel illetik. Ez az a változat, amely Hargittai könyvének a címében is szerepel, és amelytől Kaszanyiczki óva inti az olvasókat: „A Standard verzió, amelyet Learning változat néven az oktatási intézmények szinte anyagáron vásárolhattak meg, valóban csak tanulmányozásra való, a Visual Basic 5.0 igazi újdonságai hiányoznak belőle.”

A pontosabb információ úgy szól, hogy a „Learning” változatban lévő két CD közül az egyik maga a program, a másik egy videofájlokat tartalmazó oktatólemez „Learn VB Now” címmel. A „profi” változatban (Professional Edition) ehhez járul még egy harmadik CD is, válogatással a Microsoft Development Network anyagaiból. A harmadik, „vállalati” (Enterprise) változatot adják természetesen a legdrágábban: ezen

már nagyméretű, sok fejlesztő bevonásával készülő projektekhez is alkalmas. Benne van mindaz, amit a profi változat tud, plusz további eszközök a csoportos fejlesztés egyszerűsítésére és biztonságosabbá tételére. Csak ez a változat tartalmazza a Microsoft SourceSafe nevű programját, amelynek a Visual Basicbe való beépítése arra való, hogy kézben lehessen tartani, adminisztrálni lehessen az integrált programfejlesztés menetét és a forráskód kezelését.

Félreértések elkerülése végett: Kaszanyiczki könyve a középső változat, a „Professional Edition” lehetőségeit ecseteli, és Hargittai könyve is többet nyújt (ha a CD melléklet anyagát is figyelembe vesszük), mint a címben szereplő Learning Edition!

A natív kód sem csinál csodát

Az 5.0 változat legnagyobb újdonsága (de csak a „profi” változatban!), hogy el tudja kerülni az interpretatív jellegű futtatást. Azelőtt a Visual Basic változatok futtatásához mindig szükség volt egy ilyesféle elnevezésű értelmező programkönyvtárra: VBRUN300.DLL, VB40016.DLL vagy VB40032.DLL. Ezek mérete egyre nagyobb lett, a 3.0 változatnál még csak 400 KB, a 4.0-ban a 16 bites már ennek közel kétszerese, a 32 bites pedig már meghaladta a 900 KB-ot. Maga a VB fordító csak egy viszonylag könnyen és gyorsan interpretálható, ún. P-kódot állított elő a végrehajtandó programból. (Igaz, az ilyen futtatható P-kódok mérete kicsi volt, és több programhoz sem kellett több példány az értelmező DLL-ből.)

Az 5.0 változat profi változatának fordítója már létre tud hozni ún. „natív kódokat”, vagyis futtatható EXE fájlokat, sőt optimalizálni is tud, akár a méret, akár a futási idő csökkentése kedvéért. Meglepő módon azonban nem olyan egyértelmű, hogy érdemes-e mindig ezt a lehetőséget választani. Először is nagyobb lesz a natív kódú alkalmazások mérete, mert ahol eddig csak hivatkozás volt az eljárástárban található eljárásokra, ott most befordítódik maga az eljárás a programba. Másodszor, jelentős sebességnövekedést is csak az olyan programokban várhatunk, amelyekben sok a számítás-igényes elemi művelet (bit- és bájtűveletek, összehasonlítások). Amint egy program sokat használja a Windows API szolgáltatásait, vagy a karakterláncok kezelésére szolgáló komponens objektum modell (COM) metódusokat, illetve gyakran fordul a veremhez, a natív kód előnyei már minimálisra csökkenhetnek.

Kaszanyiczki László:

Visual Basic 5.0

Felhasználóknak

LSI Oktatóközpont, 1998

420 oldal, ár megjelölése nélkül

Hargittai Péter

Visual Basic 5.0

Learning Edition

LSI Oktatóközpont, 1998

418 oldal, CD-melléklettel,

ár megjelölése nélkül

Open Systems '98

Válogatás az Infopen nyílt rendszeres cikkeiből



1998 végén könyv alakban is megjelenik az Infopen magazin utóbbi számainak legjobb és legidőállóbb cikkeiből készült válogatás, köztük magyar nyelven egyedülállónak számító tanulmányok a Javáról, CORBA-ról és más nyílt rendszeres technológiákról; interjúk a szakma meghatározó informatikai vezetőivel; esettanulmányok és könyvismertetések. A könyv CD-mellékletet is tartalmaz majd többek között az Infopen magazin és az infopen.x hírlevél teljes archívumával.

Önnek azonban már nem kell az év végéig várnia!

Az Új Alaplap 1998. őszi számainak CD-melléklete Infopen rovattal jelenik meg, ahova szintén felkerülnek a válogatásban található cikkek. A szeptemberi Új Alaplap CD-n a következő Infopen-cikkek lesznek megtalálhatók:

- Enterprise JavaBeans, white paper, fordítás, Infopen 1998. április
- A 64 bites számítástechnika előnyei, white paper, fordítás és kiegészítő boxok, Infopen 1998. április
- Az extranet forradalmasítja az üzletet, Novell/Gartner Group tanulmány, fordítás kiegészítő boxokkal, Infopen 1998. március
- A Java platform, white paper, fordítás, Infopen 1998. február
- ATM technológia használata lokális és nagy távolságú hálózatokon, Arató András Networkshop-előadása alapján készült cikk, Infopen 1998. február
- Technológiaváltás a PC-piacon, Széll Zoltán, Infopen 1998. február
- Az Ethernet újjászületése, Tiszai Tamás, Infopen, 1997. december
- Mérföldkövek az NIIF-hálózat fejlesztésében, interjú Csaba Lászlóval, Infopen 1997. szeptember
- Az IPv6 hálózati protokoll, Szígyeti Sz., Mohácsi J., Máray T. Networkshop-előadása alapján készült cikk, Infopen 1997. július–augusztus
- CORBA alapok, Kovács András, Infopen 1997. július–augusztus
- Mikroprocesszor- és számítógép-architektúrák, Széll Zoltán, Infopen 1997. július–augusztus
- Elosztott rendszerek programozása Javában, Kiss István, Infopen 1997. április
- Hálózatkezelés Javában, Kiss István, Infopen 1997. március
- Objektumrelációs adatbáziskezelés, Balogh Kálmán, Infopen 1997. január–február
- Adatbázis-kezelés Javában, Kiss István, Infopen 1996. november
- Biztonságos-e a Java?, Kiss István, az Infopen 1996. augusztus–szeptemberi számában megjelent cikk bővített változata
- A Java testvérkéje: JavaScript, Kiss István, példákkal megtűzdelt bevezető a JavaScript parancsnyelvbe, az Infopen 1996. június–júliusi számában megjelent cikk bővített, csinosított változata
- A CORBA-technológia, Juhász István, Infopen 1996. június–július
- A Java programozási nyelv rejtelmek, Kiss István, bevezető a Java programozásba, elsősorban C vagy C++ programozóknak, az Infopen 1996. március–április–májusi számaiban megjelent cikkek összevont, bővített, csinosított változata
- A Java programozási nyelv, Kiss István, átfogó ismertető a Java programozási nyelvről, tulajdonságairól, képességeiről, az Infopen 1996. februári számában megjelent cikk bővített, csinosított változata

Az Új Alaplap CD az újsággal együtt 588 Ft-os áron lesz megvásárolható az újságárusoknál, illetve az Új Alaplap kiadójában (tel.: 356-1182, fax: 375-3539, e-mail: alaplap@mail.datanet.hu).

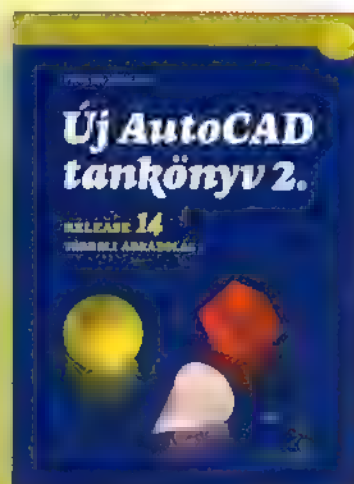
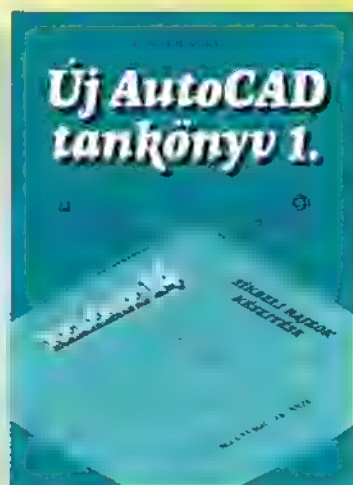
Az Open Systems '98 könyv beszerezhető lesz a számítástechnikai könyveket forgalmazó könyvesboltokban, vagy megrendelhető postai utánvétellel az Openinfo Kiadónál (tel.: 209-5400/123, fax: 466-7503, e-mail: sales@infopen.hu).

Viszonteladói értékesítés: Kiskapu Kft.

(tel./fax: 303-9119, 303-1619, nagykereskedelmi raktár: 1081 Budapest, Népszínház u. 29.).



*Ha kéri,
elküldjük
ingyenes
katalógusunkat.*



COMPUTERBOOKS

1126 Bp., Tartsay Vilmos u. 12.
Levélcím: 1253 Budapest, Pf. 71.
Telefon/Fax: 3751-564, 3753-591
Faxbank: 2333666/1456#
Email: info@computerbooks.hu



1138 Budapest, Cserhalom út 4.
Telefon: 270-6227, 270-6235
Telefon/Fax: 270-5093

Számítástechnikai rendszerek komplett hálózatának tervezése és kivitelezése

ADATHÁLÓZATOK, ERŐSÁRAMÚ HÁLÓZATOK, HÍRKÖZLŐ HÁLÓZATOK

- **BIT** HÁLÓZATI ELEMÉK
- ÖSSZEKÖTŐ KÁBELEK
- RACKSZEKRENYEK, RACKSZERELVÉNYEK
- HÁLÓZATFELÜGYELŐ RENDSZER
- ERŐSÁRAMÚ ELOSZTÓSZEKRENYEK
- TÚLFESZÜLTÉG-LEVEZETŐK HÁLÓZATOKHOZ, GÉPEKHEZ
- SZÁMÍTÓGÉPEK

Rövid kivitelezési határidő, hároméves garancia!

MASTERS of the Net

INFORMÁCIÓKÉRÉS: 12 ▲

Ready COMPUTERS

Bp. V. ker. Vadasz utca 36
11-P 8-30 - 00-20 900-1300
Tel: 331-0518 Fax 311-8671
Arlista 2-333-666/1310#
<http://www.ready.hu>

Komplett konfigurációk

IP200MMX/16MB/2.1GB/24xCD/SB Pro	136.260
IP233MMX/32MB/3.2GB/24xCD/SB16	168.560
IP-2 233/64MB/3.2GB/24xCD/SB32	249.500

+ minden géphez: ház, 14"/17" CSVGA,
1/4MB VGA, billentyűzet, eger, 1.44FDD

Áraink ÁFA-s árak!

586 VIA alaplap/Pro komp. hangk.	8.750 / 2.250 Ft
AcortX 512/SB 16	16.250 / 7.250 Ft
Intel P166MMX/IBM 200MX	21.000 / 14.250 Ft
Intel P-2 233/266	41.875 / 51.625 Ft
2.1GB/3.2GB Seagate	28.750 / 31.500 Ft
2.1GB/3.2GB SE Quantum	30.375 / 33.750 Ft
Monster 3dfx 4MB/II 12MB	23.750 / 61.250 Ft
Trident PCI/SB AWE 64	5.000 / 14.250 Ft
24x Sony/34x Asus	14.000 / 17.875 Ft

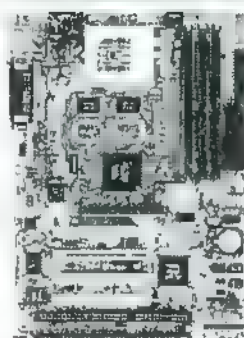
Erdeklődjön viszonteladói árainkról: (30) 413 453
Ready Compker Kft.
Az ár, ellátás joga fenntartjuk

INFORMÁCIÓKÉRÉS: 14 ▲



Mainboard Series

Features PA 2012



- ATX
- VIA Apollo Vp3 Chipset
- AGP slot
- EDO/SDRAM/ECC memória slot
- Intel MMX 166-233 MHz
- AMD K6 PR166-300

Minden **E1** alaplap egy regisztrációs kártyát tartalmaz, amellyel Ön jogosultságot szerez:



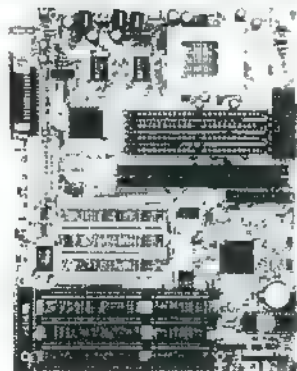
- garanciára
- IstMainboard BoardWatch elektronikus újságra a legfrissebb technikai információkkal
- BIOS frissítésre az alaplaphoz

Features VB 601

- 100 MHz FSB támogatás
- ATX
- Intel 440BX AGPset
- SDRAM/ECC memória slot
- Intel PII 233-450 Mhz
- AGP slot



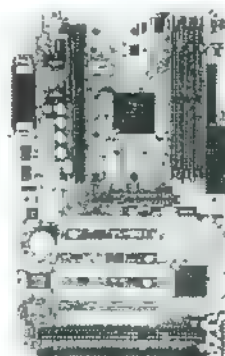
Features PT 2011



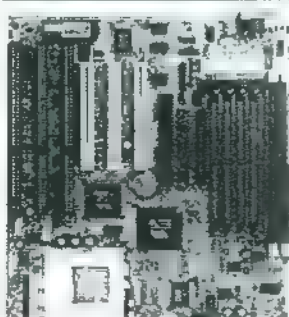
- ATX
- Intel 430TX Chipset
- EDO/SDRAM memória slot
- Intel MMX 166-233 MHz
- AMD K6 PR166-300

Features VL 601

- ATX
- Intel 440LX AGPset
- EDO/SDRAM/ECC memória slot
- Intel PII 233-333 Mhz
- AGP slot



Features VA 503



- 100 Mhz, FSB támogatás
- Baby AT
- VIA Apollo MVP3 Chipset
- EDO/SDRAM/ECC memória slot
- AGP slot
- Intel MMX 166-233 Mhz
- AMD K6 PR166-300

BEVEZETŐ ÁRAK!

További kérdéseikkel forduljanak bizalommal
Kis Ildikó kollégánkhoz
12 mellék



VAR COMPUTER

üzlet nyitvatartás
11-P 8.30-18.00

E-mail: var@var.hu
Web: www.var.hu

1149 Budapest Fogarasi út 11/a; Tel: 22-22-827; Fax: 36-32-781

INFORMÁCIÓKÉRÉS: 18 ▲

A vágólaptól az ActiveX-ig

Az együtt futó alkalmazások közötti adatsere azzal kezdődött, hogy a vágólappra kitett információt egy másik program át tudta venni. A folytatást a dinamikus adatsere (DDE) lehetőségének a megteremtése jelentette. Ez már azt is megengedte, hogy párbeszédet lehessen folytatni két alkalmazás (két ablak) között. A Visual Basicen belül például — ha előzőleg a megfelelő kapcsolatfelvétel, a „csatolás” megtörtént — a szöveglapok, grafikus ablakok vagy feliratok (címkék) mint kliensek kérhetnek információkat a munkalapoktól (űrlapoktól) vagy eredeti nevükön formoktól. A kiszolgálást az ún. vezérlők biztosítják tulajdonságaikon keresztül. A DDE tehát úgy fogható fel, mint egy automatizált vágólapp.

Lényeges különbség még az egyszerű vágólapp és a DDE között, hogy DDE kapcsolatból egyszerre több is élhet. A párbeszéd végrehajtását maga az operációs rendszer, a Windows teszi lehetővé, és az információt szolgáltató szerver szerepét éppúgy, mint az információt kérő kliensét egyszerre több alkalmazás is betöltheti. Egy Visual Basic projekten belül például a Word továbbíthat olyan információt a Windowsnak, amelyet az Exceltől vesz át. A „beszélgetés” tárgyát a fájlnev fejezi ki, az Excel esetében ez lehet például egy cella vagy egy kijelölt tartomány.

A következő lépés az OLE kapcsolat megteremtése volt. Ennek az a lényege, hogy egyszerű adatsere helyett pontosan olyan formában tudja megjeleníteni az átvitt információt, ahogy az az eredeti alkalmazásban kinézett. Egy Visual Basic alkalmazásban tehát például „nyitni tudunk egy ablakot”, és azon

keresztül belelátunk egy másik alkalmazásba, mondjuk láthatóvá tesszük egy Excel táblázat valamelyik részét.

Az OLE2 alkalmazás még tovább ment: lehetővé tette, hogy magukat a megjelenített adatokat a befogadó alkalmazáson belül lehessen manipulálni, más szóval egy fedél alatt lehessen integrálni több alkalmazást. Ideiglenesen tehát az új alkalmazás veszi át a vezényletet a beágyazott adatokon.

Ennél is többet lehet elérni az ún. OLE automatizmus segítségével, amely elvi lehetőségként ugyan már a 4.0 változatban megjelent, lassúsága miatt azonban gyakorlati felhasználásra a legprimitívebb mintaprogramokkal is alkalmatlan volt. Az OLE automatizmus esetében a továbblépést az jelenti, hogy bizonyos funkcionálisan önállóan működő önálló részeket (magyarán: objektumokat, az objektumok definícióival, metódusaival és tulajdonságaival együtt) más alkalmazásból szűrőstülbőristül kölcsön lehet venni, hozzá lehet csapni a kölcsönvevő projekt eszköztárhoz. Itt tehát nemcsak egy „adatablak” nyílik a másik alkalmazás valamelyik adatterületére, hanem a működését biztosító eszközöket is kölcsön lehet venni a másiktól. A kölcsönvételhez természetesen arra is szükség van, hogy ezek az objektumok kölcsönadható formában legyenek elkészítve, és így legyenek elkészítve a „kölcsönzőben”, egy ún. objektumtárban.

Az ActiveX lényegében ugyanazt a funkciót tölti be, mint az OLE elemek, csak éppen a Microsoft ízlése szerint szabványosítva, a COM (komponens objektum modell) követelményeinek megfelelően. Az OLE vezérlőelemek így átlényegülnek ActiveX vezérlőelemekké, az OLE DLL könyvtárak ActiveX DLL könyvtárakká, az OLE objektumok pedig ActiveX EXE alkalmazásokká. Újabb találmány viszont az ún. ActiveX Document, amelynek segítségével már internetes alkalmazásokat is létre lehet hozni: projektünk formjait (munkalapjait, űrlapjait) ActiveX dokumentumokként lehet továbbítani. Az egész folyamatot természetesen aktívan támogatja egy jóindulatú varázsló, az ActiveX Document Migration Wizard.

Kaszanyiczki László könyve jól kiemeli és koncentráltan adja tovább a lényeget. A menük részletes, gazdagon illusztrált ismertetésétől a VB 5 nyelvnek rendszerezett ismertetésén keresztül jut el a programozás leglényesebb és legkényesebb kérdéseinek tárgyalásáig. Apróság, hogy az objektumok tárgyalása miatt ékelődött bele egy másik témakör közepébe. A tárgy-



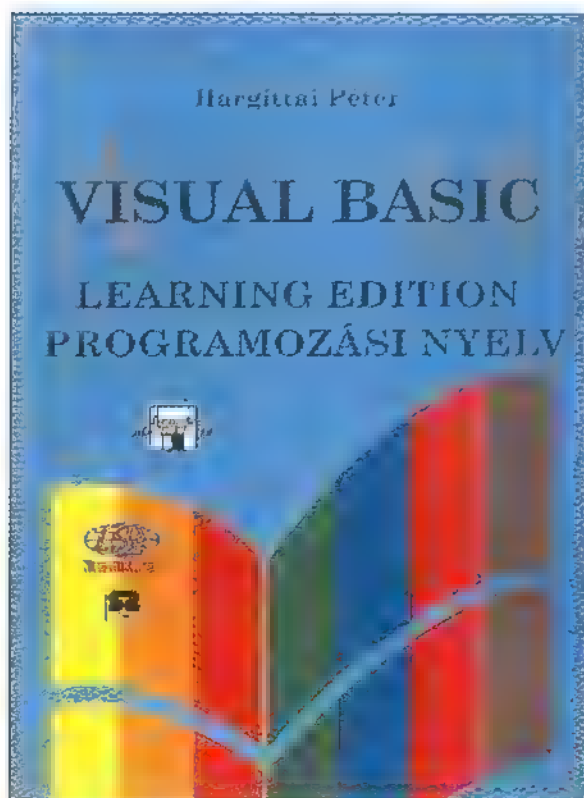
mutató is hiányos helyenként, és tele van 1-2 oldalas elcsúszásokkal.

Referencia-kézikönyv

Nem értékítélet, hogy Hargittai Péter könyvéről sokkal kevesebb a mondani-valóm. A maga nemében hasznos mű, de a referenciaszerű anyagból nehéz kihámozni a lényeget. (Nem is arra való persze, hanem a részletek megismertetésére.) Valami rövid eligazítást mégis hozzáfűzhetett volna a szerző másféle rendszerezésben is: például tematikus áttekintést egyes nagyobb témakörökről, így az adatbáziskezelésről, a kommunikációról vagy az objektumok hierarchikus összefüggéseiről. Hiányzik a könyvből a tájékoztatás az alapobjektumokról. Igaz, hogy a Microsoft elképzelései sem alakultak még ki elég világosan ebben a kérdésben, de legalább össze lehetett volna foglalni azokat a változásokat, amelyek a 4.0-ról az 5.0-ra való áttérés során történtek (az általános objektumok legtöbbszörének „átminősítése” Global objektum tulajdonságúvá stb.).

Igen hasznos, hogy a szerző a különböző elemek (objektumok, függvények, utasítások, metódusok, események, tulajdonságok) betűrendes tárgyalása előtt átfogó listát ad róluk, és félkövér szedéssel kiemeli az 5.0 változatban újonnan megjelent vagy lényegesen megváltozott nyelvi elemeket. Sajnálattal viszont, hogy a könyvnek a CD-re szorult, kéziratként tökéletesen kidolgozott folytatásából legalább ezek a tartalomjegyzék-szerű rövid listák nem kerültek be a nyomtatott anyagba. Akkor az olvasónak legalább áttekintése lenne a VB 5.0 által felölelt egész fogalomköréről.

Vargha Dénes



„Mától a gépeden...”

DB2 for Windows NT

Az IBM az utóbbi időben egyre több figyelmet szentel a Windows NT-nek, mert — mint az egyik sajtótájékoztató anyagában olvasható — „A teljes NT-piacon az operációs rendszerek nyeresége a lehetőségek(nek) csupán 10%-áig terjed. Az IBM most a többi 90%-ra koncentrál, amely hardverből, szoftverből, valamint szolgáltatásokból áll.” Az NT-s IBM-stratégia ismeretében nem meglepő tehát, hogy megjelent a színen a DB2 Windows NT alatt, illetve a Whitehorn szerzőpárosnak ezzel a témával foglalkozó könyve, a DB2 for Windows NT.

Mark Whitehorn, igazi SQL-hívő, aki a University of Dundee-n a relációs adatbáziskezelés elméletét és gyakorlatát tanítja, másfél évtizednyi tapasztalat alapján. Havi rovata is van a Personal Computer World angol számítástechnikai lapban (az adatbázisokról), és egy másik szerzővel közösen jelent már meg könyve szintén a Springernél (természetesen a relációs adatbázisokról). Mary Whitehorn viszont arról szokott írni a számítógépes szaksajtóban, hogy miként oldjunk meg bizonyos gyakorlati problémákat („hogyan installáljuk

X szofvert 12 lépésben”), tehát egyáltalán nem meglepő, hogy ez a szerzőpáros valóban könnyen áttekinthető, olvasmányos és hasznos könyvet tudott írni. Olyan művet, amelyet az ember például akkor forgat haszonnal, ha reggel bemegy a munkahelyére, és egy cédulát talál az asztalán, amely szerint „a gépeden máától DB2 fut. Használd egészséggel”. Ilyenkor persze meglehetnénk, hogy a szoftverrel együtt érkező, rendszerint meglehetősen vaskos kézikönyveket kezdjük el böngészni, ám egy ilyenben tényleg mindennek benne kell lennie, és ez könnyen oda vezethet, hogy az ember nem igazán boldogul vele. Esetleg képtelen ugyanis eldönteni, hogy melyik információ igazán fontos számára az első lépésben, és melyik nem az.

Nos, éppen ezt az igényt figyelembe véve írták a DB2 for Windows NT című könyvet. Ez ugyanis azzal kezd, hogy elmagyarázza azt (amit én eddig nem ismertem), hogy mi is az a DB2. Hogy ez tulajdonképpen egy DBMS, azaz adatbáziskezelő rendszer (database management system), amely szükség esetén olyan nagy mennyiségű adattal

dolgozik, amekkorára például egy cégnek van szüksége. De a DB2 bizonyos szempontból több is ennél, RDBMS, azaz „relational”. A DB2 esetében nem egy gépen található meg minden, hanem kliens-szerver viszony áll fenn az egyes elemek között.

Ahhoz persze, hogy a DB2 ezen „GUI-verziója” egyáltalán elinduljon, szükséges bizonyos hardverkövetelményeknek is eleget tenni, miként azt a bevezetésből megtudhatjuk. És itt derül ki az is, hogy melyek azok a minimális előismeretek, amelyek nélkül nem érdemes továbblépni (vagy ha nem vagyunk tisztában az alapokkal, merre érdemes elindulni).

Ezt követően az installálással ismerkedhetünk meg lépésről lépésre, illetve képernyőről képernyőre, majd pedig a szerzőpáros által legfontosabbnak tartott elemek bemutatása következik (amilyen például a „Control Center”). Megtudhatjuk azt is, hogy miként lehet installálni egy ún. „front-end tool”-t az adatok kezeléséhez, és milyen rendszeradminisztrátori teendőket kell ellátnunk a DB2 működtetéséhez, milyen eszközök segítségével tudjuk figyelemmel kísérni a rendszer működését... és így tovább. Aki figyelmesen követi a könyv utasításait, az ettől még nem lesz DB2-szakértő, de a gyakorlati oldalt tekintve tájékozottan fog felállni a könyv mellől. Hasznos lenne, ha a könyv magyar fordításban is megjelenne.

Galántai Zoltán



K&Szo Kft

1055 Budapest V., Falk Miksa u. 6.

Telefon: 332-8717

Fax: 302-5136

E-mail: sales@keszo.com

Web: www.keszo.com

National Geographics komplett gyűjtemény CD-n	52.000	MS Office 97 magyar / upg.	112.800 / 47.200
Windows Commander 3.5 16/32bit (magyarul is)	8.800	MS Office 97 magyar prof. / upg.	135.800 / 70.000
FAR 1.51 / RAR 2.02 / ARJ 2.6	8.000 / 8.000 / 16.000	WinFAX Pro 8.0 NT, Win95 / upg.	31.000 / 16.000
Winzip 6.3 / Pkzip 2.04g	14.000 / 17.000	Partition Magic 3 (particionálás adatvesztés nélkül)	23.000
Windows Commander 3.51	8.800	Visio 5.0 Win95/NT Standard / upg.	54.000 / 36.000
Teleport Pro (weblapletöltő)	22.000	Visio 5.0 Professional Win95/NT / upg.	112.000 / 52.000
F-Prot Professional	43.000	Visio 5.0 Technical Win95/NT / upg.	112.000 / 52.000
Clarion Developer 4.0 / upgrade	126.000 / 58.000	Photoshop 4.0 Win95/NT / upg.	184.000 / 65.000
Hot Metal Pro 4.0	36.000	Photoshop 4.0 Win95/NT magyar! / upg.	184.000 / 69.000
MS Frontpage 98	33.600	NT 4.0 Server / WKS Resource Kit	34.000 / 16.000
MS Project 98 / upgrade	112.000 / 44.900	Win 95 Resource Kit / Office 97 Res. Kit	9.800 / 14.000
Norton Utilities 3.0	28.000	Norton Commander 1.2 Win95/NT / upg.	20.000 / 11.000
Norton Antivirus 4.0 angol vagy magyar	19.600	Adobe Acrobat / Corel ArtShow 7	76.000 / 14.400
System Commander 4.x Deluxe	30.000	Multikey 3.5 / upgrade	3.600 / 2.000
Norton Uninstaller	16.000	NT KEY 4.0 / upgrade előző verziókról	10.000 / 6.000
MathCAD 7.0 Professional	128.000	Adobe Illustrator 7.0 / upg.	125.000 / 45.000
Procomm 4.5 Win95/NT Internet, fax, modem, rc.	51.000	Siearra Home +Garden	24.000
Unicode TrueType 100 betűtípus	3.900	QuarkXPress 4.0 Win95/NT / PowerMAC	244.000
Drivelmage (FAT16/32, HPFS, NTFS)	24.000	Helyes-e? for QuarkXpress 4.0	59.000
Adobe Type Manager 4.0 deluxe for NT	25.000		

Áraink áfa nélkül értendőek!

Novell®

Ha hálózhat, akkor

ELŐFIZETÉS

Az 1998/..... számtól kezdődően előfizetem

az Új Alaplap című CD-mellékletes havi számítástechnikai folyóiratot

..... példányban ☐ 1 évre, ☐ 1/2 évre.

Az éves előfizetési díj: 5880,- Ft (Ez az összeg az áfát is tartalmazza.)

☐ Számlát kérek (banki átutalással fizetek).

☐ Befizetési csekket kérek.

Név:

(Cég:)

Cím:

Irányítószám, helység:

Dátum:

.....
/aláírás/

APRÓHIRDETÉS

Kérem, hogy az Új Alaplap következő számának Mikrobazár rovatában az alábbi szövegű apróhirdetést jelentessék meg. (A túloldalon ismertetett feltételeket tudomásul veszem.)

.....
.....
.....
.....
.....
.....
.....
.....
.....

(Maximális terjedelem 300 betű.)

INFORMÁCIÓKÉRÉS

Az Új Alaplap mostani számában megjelent hirdetések közül az általam itt megjelölt kódszámúakhoz részletesebb információt kérek a hirdető cégektől.

01	02	03	04	05	06	07	08	09	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96

Belföldön
díjmentesen is
feladható

ÚJ ALAPLAP

**VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest**



Feladáskor kérjük bérmentesíteni!

FELADÓ:

Név:

Cím:

Helység:

Irányítószám:

Telefon:

☐ A hirdetés egyéni és egyedi jellegű, ezért kérem ingyenes megjelentetését. Kijelentem, hogy annak tartalma nem sérti senki szerzői jogát.

☐ A hirdetés kereskedelmi célt szolgál. Mellékelem a soronként (60 karakterenként) 300 forintnak megfelelő összeg átutalásáról az igazoló szelvény másolatát. A címzett: Új Alaplap, 1539 Budapest, Pf. 571, illetve átutalásnál az OTP 11706016-20788599 számlaszámra.

/aláírás/

Bélyeg
helye

ÚJ ALAPLAP

**VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest**



FELADÓ:

Feladáskor kérjük bérmentesíteni!

a) EGYÉNI

Név:

Cím:

Helység, ir.sz.:

b) CÉGES

Név:

Cég:

Cím:

Helység, ir.sz.:

Telefon:

/aláírás/

Bélyeg
helye

ÚJ ALAPLAP

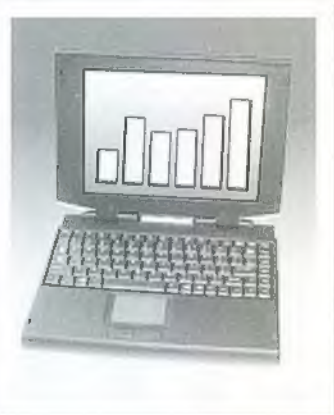
**VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest**



KAO ... a tökéletes memória
Media from the Surface Scientists

**A SZÁMÍTÁSTECHNIKA TELJES KÍNÁLATA.
HITEL ÜGYINTÉZÉS HELYBEN, KEZES NÉLKÜL IS!**

QUERTY
COMPUTER
Alapítva 1984-ben



**Internet akár havi
alapdíj nélkül is!**

EPSON

Nyomtatók teljes választéka,
tartozékok, kiegészítők.
Calcomp plotterek.
Portocom, Compaq, Toshiba
notebook számítógépek.
UMAX, EPSON, GENIUS szkennerek.
EPSON, OLYMPUS, AGFA digitális
fényképező gépek.
SAMSUNG monitorok teljes választéka.
DTP-rendszerek.
Multimédia eszközök, CD-írás.
Modemek viszonteladóknak is.
GSM-adatátvitel.
ISDN kapcsolat, routerek és
hálózati konfigurálás.
Szoftverek, tartozékok, kiegészítő
eszközök, szakkönyvek széles
választékával és tanácsadással várjuk!

COMPUTER SZAKÜZLET 1111 Bp., Bartók Béla út 14.
Telefon: 466-9377 Fax: 385-2687

EPSON SZAKÜZLET 1114 Bp., Bartók Béla út 9.
Telefon: 466-5419 Fax: 385-2687

Faxinfo árlistákkal: 466-8292 Internet: <http://www.querty.hu> nyitva: hétfő - péntek 10-18 óráig

INFORMÁCIÓKÉRÉS: 13 ▲

professzionális
szinvonalú,
biztonságos
inter**net**
megoldások



telnet Magyarország

1136 Budapest,
Hollán Ernő u. 9.
telefon: 302-4781
e-mail: info@telnet.hu

<http://www.telnet.hu>

INFORMÁCIÓKÉRÉS: 17 ▲

10 ÉVES

A MAGYAR CD-GYÁRTÁS...



KOMPAKTLEMEZ
KOMPAKT TECHNOLÓGIA
KOMPAKT SZOLGÁLTATÁS



VTCD VIDEOTON

Tel.: (06-22) 329-132
Fax: (06-22) 329-133
E-mail: vtcd@mail.datanet.hu
8001 Székesfehérvár Pf.: 175.

Tekintse meg internetoldalunkat is: <http://www.vtcd.hu>

